Experiences in Applying a Mobile Service Platform Across Different Business Domains

Anssi Karhinen, Oskari Koskimies and Jukka K. Nurminen

Nokia Research Center

Abstract. The purpose of this report is to share the experiences and lessons learned in fast-paced mobile service research. We describe on the evolutionary steps of the mobile service platform that has been used to field different mobile service concepts in varying business domains. The platform was developed to solve issues of rapid deployment in different field force applications for data gathering and task management. In this report we focus on the changes needed when the platform was used to develop a mobile service for fleet management and automatic data gathering in the road-maintenance domain[1].

1 Introduction

Increasing capabilities of cellular mobile devices open possibilities for developing mobile applications and services in many different business domains. Better UI, larger memory, integrated peripherals like camera and voice recorder, and wirelessly connected peripherals like GPS and bar code readers make it possible to use generic mobile phones for applications that used to require custom made, integrated systems.
While the development of mobile applications for professional use has similarities to the development of mainstream IT solutions like web based services it also has important differences. Different business domains have also fundamental differences in their requirements for mobile solutions. In this study we focus on some of those differences, how they materialize in a development project, analyze their causes, and suggest ways how to deal with them.
We have focused our research on mobile service solutions for different business domains. One of the goals has been to study the benefits of software as a service paradigm [4] that has gained a big momentum in the fixed internet based enterprise solutions.

For the empirical part of our research we have used participant observation [9] methodology where the researcher takes part in the activity, records the experiences and, after reflection, draws conclusions from this. The weakness of this approach compared to unengaged observation and analysis is that personal involvement may bias the results. The validity and generality of the results derived from the development of a single application has to be taken with care. However, as observed

[1] The authors wish to thank Henry Tennberg and YIT Infra Services Road Maintenance for their advice and support in trialing mobile services in the road-maintenance domain.

by [2] collecting such practical experiences and trying to generalize the observations from those is one of the few practical ways to understand real-world software engineering.

Development of the mobile service implementations in such environment had requirements for extreme agility, low overhead, rapid prototyping capability and high enough quality that the services could be deployed in a commercial environment for trial use.

By documenting and sharing the experiences in this challenging environment for mobile software development we hope to give some perspective to other developers to mirror their own experiences. In this paper we have focused on the special requirements for the mobile software set by our research method.

The concrete trial case in this paper focuses on the development of a mobile service solution for road maintenance. Efficient execution of the road maintenance operations is of importance for the traffic safety and fluency as well as for the road infrastructure cost. The trial application we developed for the road maintenance unit of YIT group [15] allows the company to track its road maintenance tasks, collect history data, create reports, and monitor the cost of its activities.

2 Related Research

Other researchers have studied mobile field force applications in different domains. For instance, [7] discuss the challenges of applying IT technology to the construction domain. [11] analyzes the use of a mobile solution by healthcare workers. Their focus is on the user acceptance of the applications while this paper focuses more on the development aspects of the mobile solution. Naturally the needs of different domains are also distinct.

Roadex research program [12] investigated the use of mobile technology for road maintenance. It presents a set of interesting use cases, some of which could be covered as an extension of the present work.

A growing number of mobile applications and services are available. Road maintenance solutions are available at least from [5] and [13]. In comparison to such companies our target was not primarily to create a commercially successful solution. Instead we wanted to focus on creating a better understanding of the field force services domain, on the challenges of developing mobile service applications, and on creating tools, technologies, and methods which allow easier development of mobile service solutions.

We are not aware of many documented industrial use cases for mobile field force applications such as task management or field inspections. There are some commercial players providing solutions in this space though. A company called iAnywhere [3] has products that extend database and web based applications to mobile devices. They use field force automation and especially solutions for infrastructure maintenance, construction and delivery as reference applications. Buildercom [1] is a small company that focuses on construction domain and has mobile inspection product for safety inspection process.

3 Mobile Service Platform

We had developed a mobile service research platform to serve as a basis for rapid service deployment for trials in different business and industry sectors. The functionality and non-functional characteristics of the platform were initially based on analysis of several application domains and several field studies of different industry segments that had potential for running their processes through a mobile service.

The goal of the domain analysis and field studies was to isolate reusable business processes or process fragments from different business domains. We called them "mini-processes". The platform was designed to be able to instantiate and host mini-processes that are customized to fit a particular domain and customer. It had mechanisms to create a service template library of reusable mini-processes.

The non-functional characteristics of the platform were dictated by the envisioned application domains, trial customers and the nature of the selected mini-processes. Capacity and scalability requirements were kept modest as the practical size of trials in the participant observation research method is rather small. Also typical solutions for scalable capacity like distributed processing add a lot of complexity to the basic design of the platform.

Reliability and fault-tolerance requirements were deemed to be rather reasonable as the trial usage of the services was not considered mission-critical at this point for the trial customers. Again the solution for high availability and reliability by replication of resources adds a lot of complexity to the platform. Instead we opted for single running service instance and periodic data replication.

The envisioned mobile service user for our research was a field worker who is connected to a business process through a mobile service. He can receive, view, edit and send the business documents belonging to the process and generate additional data using different sensors attached to his mobile device. Examples of mini-processes that we considered:

1. Task management

Task management is a generic process that is common to many business domains that operate a field force. Typical examples are different repair and maintenance services. A manager can assign different tasks to the field workers and track them. The field workers are able to acknowledge the task and execute the related domain specific processes.

2. Inspection

Inspection is a form of mobile data gathering where a field worker is assigned a task to collect specific data from a given location. Typical domains for inspection process are building maintenance and construction and supply chain inspections.

Common requirement for all of the mini-processes that we were planning to implement was that the field workers must be able to execute a part of the process also when outside network coverage. Thus we needed a mechanism for asynchronous and robust messaging and a client platform that is capable to execute a part of the logic of the mini-process.

Given these requirements our platform was built on top an in-house developed, robust and reliable mobile asynchronous messaging backbone. It also provided a mechanism for provisioning the client software and configurations.

The client software was built on top of MIDP 2.0 Java standard and XForms technology that made it possible to easily deploy customized user interfaces for different requirements and also handle user interaction logic locally without the need for on-line server connection.

Other technologies that were utilized to increase the agility and flexibility of the platform were Business Process Execution Language (BPEL) [8] and extensive usage of XML based data presentations.

4 Road Maintenance Solution

When we started to work on the road maintenance domain we chose the approach of customer-driven innovation. E.g. [7] indicate that without properly understanding the end-user needs it is easy to create a solution that on the surface is perfectly adequate but which fails to get customer acceptance. In comparison to office applications the need for customer involvement in field force solutions is even bigger since few developers have direct hands-on experience with field-force work.

A part of the customer-driven approach is incremental development using the customer feedback as a driver. Piloting with early beta versions and gradually improving the solution with the feedback are important tools. Rapidly reaching the pilot level was thus one of the targets.

We worked together with the road maintenance division of the Finnish company YIT group to define how the mobile service platform would be applied to the road maintenance domain. We started the work with a user research activity where we shadowed three candidate users in order to understand how they worked. The user research material and discussions with the managerial team was used to decide on the key functionality of the solution:

- Displaying the road address of the truck to the driver
- Tracking of trucks and ongoing tasks
- Recording the tasks and routes of each truck
- Creating and viewing location based notes
- Reporting material consumption
- Creating summaries of maintenance history

The solution consisted of a mobile form used by truck drivers to report on their tasks, and a web user interface used by foremen to monitor the tasks and to create reports. From truck driver point of view, the basic scenario was as follows:

1. The driver opens the road maintenance form.
2. The driver selects what type of task he is going to perform.
3. While the driver performs the task, GPS coordinates are recorded every 30 seconds and added to the form data. The latest coordinate and the corresponding address are shown to the user during task execution.
4. During the task, the driver can make notes which are recorded together with latest coordinates. This is used to make notes of fallen traffic signs and the like.
5. The driver finishes the task and presses a “Finish” button on the form.
6. The driver enters the amount of consumed materials (e.g. sand, road salt).

7. The driver submits the data, which also closes the form

In order to show the road address we needed a way to convert GPS coordinates to road addresses. We decided to use a third party service to perform the conversion. The logic of using a third party service was in line with our basic assumption to use as much existing services as possible. As such a service was available and could be modified to our needs with reasonable effort this appeared to be a reasonable choice. We could save our development resources for other tasks and geographic information systems specialists would take care of the address conversions. The phone would send its coordinates (received from a GPS device) to the third party server which would return the corresponding street/road address.

Using prototype mobile forms without backend functionality, the customer was able to provide quick feedback on the design of the mobile user interface. This was essential given our schedule – we only had two months to develop the solution. The target devices for the pilot were Nokia 6680 and E70 mobile phones running Symbian S60 operating system.

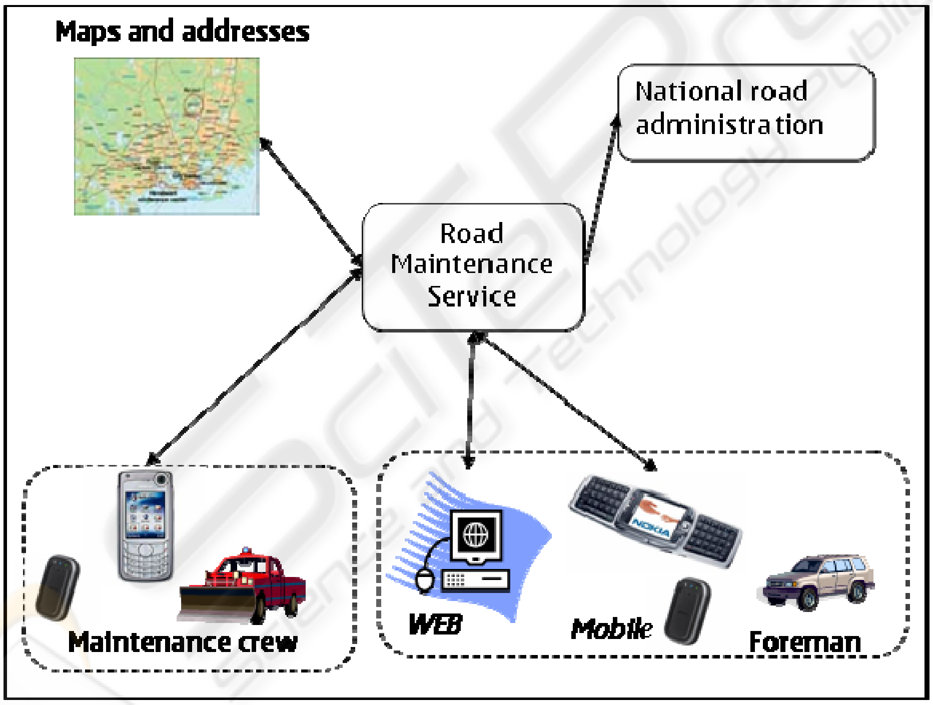


Fig. 1. Overall structure of the service.

Figure 1 shows the overall structure of the service. Mobile maintenance crews and foremen are equipped with mobile phones and GPS receivers. They continuously communicate their location to the service. Map and address translation service is used to convert the GPS coordinates to road/street addresses.

Browsing the stored information is possible via a web user interface. It is typically used by foremen to create reports and monitor the work. Finally, a subset of the data is transferred to the national road administration which is responsible for the nation's road network.

5 Applying the Mobile Service Platform to Road Maintenance

Our initial evaluation was that the road maintenance use case was fairly close to a regular form filling application such as an inspection form. The only exception was the automatic collection of GPS coordinates into the form data. However, the XForms technology used in the mobile service platform was relatively easily extended to support GPS integration. The address translation was performed by a separate GPS software component before the data was injected into the form. The address translation did introduce real-time requirements for communication because the server-side address translation has to be performed quickly in order to show the result to the user while it is still relevant. Originally the platform only supported a batch communication model – a reliable message queue – where sending a message could take an arbitrarily long time to complete. Address translation, on the other hand, required a synchronous protocol, namely Simple Object Access Protocol (SOAP).

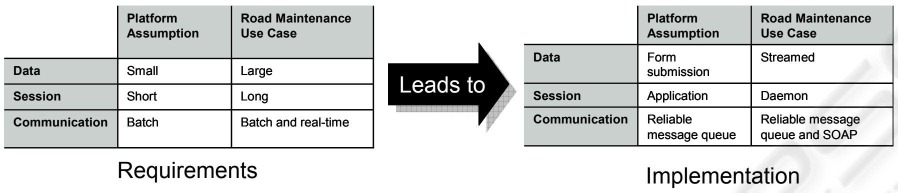
Once the pilot had been running for some time it became apparent that our initial assumption was false in two respects. Firstly, the time interval between coordinates was too long. Initially we executed the address conversion once in every 30 seconds. Once we started getting early comments from our pilot users we realized that this was too infrequent. A truck driving 80 km/h is able to move close to 700 meters in that time. Consequently the time interval was reduced to 5 seconds. This solved the accuracy problem, but increased data output by a factor of six, causing heavy load at the address translation server and vastly increased costs in both address translation (which was charged for on a per-transaction basis) and cellular data traffic. The problem was exacerbated by the use of SOAP requests for the address translation [Lesson 1]. In hindsight, performing the address conversion on the mobile device would have been a better choice [Lesson 2]. With the server-side address translation service, the system operating expenses are simply too high for full-scale commercial deployment. However, at the time when the decisions were made the service based solution seemed to be optimal.

Secondly, we had assumed that an application session, i.e. the performance of one task, would have a relatively short duration. The typical scenario was the spreading of sand on a single road. In reality, certain tasks can last a full working day. This had two serious implications:

1. Together with the shorter time interval between coordinates, day-long sessions meant that the data of a single session could not fit in phone RAM. Whereas previously all data had been submitted at the end of the session, we now had to introduce a streaming model where data was continuously submitted using small partial submissions. Fortunately XForms is expressive enough to support this.
2. The application has to be able to run in the background, like a daemon, for a long period of time and keep recording coordinates. This is difficult

because the S60 operating system will kill the road maintenance application to reclaim memory if another application with non-trivial memory requirements is started. For example, it was not possible to use a third party mobile email application at the same time as the road maintenance application. The users were essentially forced to use a dedicated phone for the road maintenance application.

The effects of the new requirements on the mobile service platform are summarized in **Fig. 2.**



	Platform Assumption	Road Maintenance Use Case
Data	Small	Large
Session	Short	Long
Communication	Batch	Batch and real-time

	Platform Assumption	Road Maintenance Use Case
Data	Form submission	Streamed
Session	Application	Daemon
Communication	Reliable message queue	Reliable message queue and SOAP

Fig. 2. Effects of new requirements on the service platform.

After handling these issues, the pilot application was working well for some time. Then a new pilot site was introduced, in a remote location where there are significant gaps in GPRS coverage. We suddenly started experiencing reliability problems with the software – crashes, hang-ups, and so on. After investigating the issue we found out that the problems were caused by the intermittent connectivity at the new pilot site. While we had taken care to support both offline and online use, we had not tested the middle ground, i.e. intermittent connectivity. The cellular network coverage is very good in Finland, and as a result our field tests never encountered a situation where the phone temporarily loses connectivity. The problem was simply that from protocol stack point of view, intermittent connectivity is different from switching between offline and online, and has to be tested and debugged separately [Lesson 3].

Interestingly, replicating the problem turned out to be challenging. The modern office building where the developers are working has very good cellular coverage. The nearby city area is also very well covered. Some experiments were made microwave ovens, kettles, and other household equipment. A kettle inside a microwave oven turned out to block enough radiation to break the data connection. It also turns out that there is significant variation between phone models; a setup which breaks the data connection for one phone model may be insufficient for another model [Lesson 4].

Meanwhile we were able to find a more professional solution from a group of antenna researchers working in our building. The radiation isolation ball made of copper (see **Fig. 3**) is used for antenna testing. It turned out to be a useful tool for us allowing easy testing to see how the application behaves when the phone moves in and out of coverage. The drawback of such a solution is that it is not possible to operate the phone while it is out of coverage. We later found out that the antenna researchers have a special room for that purpose in the basement.

For the GPS location technology we had the opposite problem. It worked well in the field but not indoors. In the early phase of the development we took some walks outside to test the application. Naturally this was very time-consuming. Furthermore, it was difficult to demonstrate the application to the customers which is important in the customer-driven development. We ended up developing a very simple GPS emulator. While it allowed most development activities to be done, the arbitrary coordinates returned by the emulator did not match the reality. A well though-out emulator should be able to return coordinates on roads only and consecutive coordinates should emulate a vehicle driving with a given speed. It should also be possible to emulate cases where the location is unknown because of missing visibility to the satellites [Lesson 5].

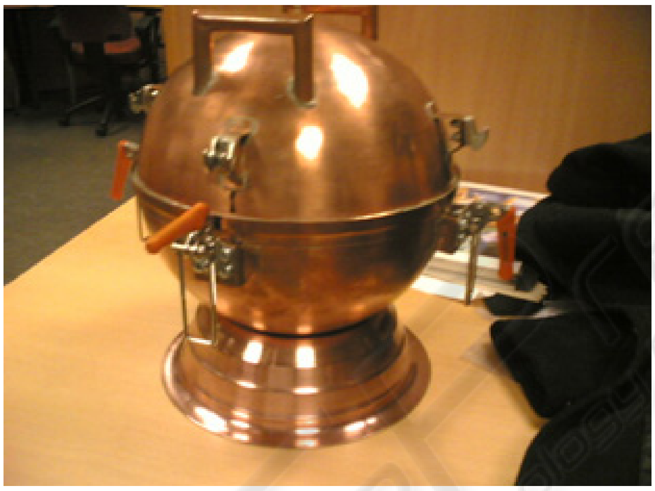


Fig. 3. A copper sphere used to test out-of-coverage situations.

While we have so far concentrated on the things that did not go as expected, because those are usually the most interesting, it is worth pointing out that one thing which is usually problematic with mobile applications was in this case relatively straightforward: the user interface.

The small screen size is frequently a problem when developing mobile applications. In this work, however, the mobile user interface did not present any major problems. Most of the tracking activities are performed automatically. The interactions that the driver has to perform are typically done before and after driving and do not require complicated user interaction. At the beginning the driver needs to select the task that he is currently doing (e.g. laying salt). When the task is completed the driver needs to report that the task is completed together with the amount of materials (e.g. salt) consumed [Lesson 6].

6 Lessons Learned

1. *SOAP is not an appropriate solution when a system has a high traffic volume e.g. in the case when the real-time requirements are high. It may still be a good option for the early version of the system for allowing faster beta*

testing. In general this lesson applies to extensive usage of XML based data presentations.

2. *When the mobile device needs to display real-time data, do not use an external service. Instead have the database and necessary processing routines in the device itself. This lesson can be generalized to service oriented architectures in general.*

3. *Plan and test the application so that it works in online, offline and intermittent connectivity scenarios. All three must be tested separately.*

4. *Specialized tools are needed to test how applications react when the phone experiences problems with cellular connectivity.*

5. *A GPS emulation tool should be part of a mobile application development tool set.*

6. *Mobile user interfaces for maintenance applications work well despite the small screen area. The needs for interaction are not very extensive. A few selections from drop down lists are enough for most tasks, and some data can even be collected automatically. However, being able to store and share relevant info in a timely fashion has great value.*

7 Conclusions

Researching the utility, user experience, perceived value and scalability of mobile service concepts requires empirical studies to be carried out in the real business environment. For this purpose we have applied customer driven innovation and participant observation methods and deployed trial configurations of various mobile service concepts in different business domains.

Fast moving service research across different business domains sets special requirements for the underlying platform components of the service implementation. The main focus is on rapid development and deployment of the services and agile adaptation to emerging requirements. To meet these challenges we have utilized technologies like XForms and BPEL that abstract the UI features and processing logic into a more flexible form compared to native application development.

The mobile service research platform was developed to meet the requirements of the initial service concepts and application domains. In particular it was targeted to services for mobile field force and certain generic processes like task management and mobile data gathering. As the concepts evolved and new business domains like road maintenance and fleet management were included in the research the platform had to evolve. In this report we described a major evolution step from the original set of requirements for slow-paced manual data gathering to fast-moving automatic data generation and two-way real-time communication.

The experiences that we wanted to share are condensed into set of learnings that arose during the refactoring process of the platform. We believe that the utilization of mobile services is still in its infancy in many business sectors and hope that our experiences will benefit the development of highly useful mobile services for many business domains with different requirements.

References

1. Buildercom Ltd, http://www.buildercom.net/
2. Fenton N.E., "Conducting and Presenting Empirical Software Engineering", Journal of Empirical Software Engineering 6(3), 195-200, 2001
3. iAnywhere Solutions Inc., http;//www.ianywhere.com
4. IDC, 2005. Software as a Service Taxonomy and Research Guide, http://www.idc.com/getdoc.jsp?containerId=33453&pageType=PRINTFRIENDLY#33453-E-0004
5. Incode Ltd. http://www.incodesoftware.com
6. Juric, M.B., Rozman, I., Brumen, B., Colnaric, M., and Herick, M., Comparison of performance of Web services, WS-Security, RMI, and RMI-SSL, Journal of Systems and Software, Volume 79, Issue 5, Quality Software, May 2006, Pages 689-700.
7. Mitchell, V., May, A., Bowden, S., and Thorpe, T. 2006. Using mobility as a conceptual framework for informing the design of mobile ICT for construction professionals. In Proceedings of the 8th Conference on Human-Computer interaction with Mobile Devices and Services (Helsinki, Finland, September 12 - 15, 2006). MobileHCI '06, vol. 159. ACM Press, New York, NY, 45-48. DOI= http://doi.acm.org/10.1145/1152215.1152225
8. OASIS Technical Committee, Web Services Business Process Execution Language (WSBPEL), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel
9. Pidd, M., Dunning-Lewis, P. 2001. Innovative research in OR/MS? European Journal of Operational Research 128, 1-13.
10. Rajlich, V. 2006. Changing the paradigm of software engineering. Commun. ACM 49, 8 (Aug. 2006), 67-70. DOI= http://doi.acm.org/10.1145/1145287.1145289
11. Sammon, M. J., Brotman, L. S., Peebles, E., and Seligmann, D. D. 2006. MACCS: enabling communications for mobile workers within healthcare environments. In Proceedings of the 8th Conference on Human-Computer interaction with Mobile Devices and Services (Helsinki, Finland, September 12 - 15, 2006). MobileHCI '06, vol. 159. ACM Press, New York, NY, 41-44. DOI= http://doi.acm.org/10.1145/1152215.1152224
12. Saarenketo, T., Monitoring, communication and information systems & tools for focusing actions: Ideas and Innovations. Final report from the Phase III subprojects 3_2 and 3_4 survey of the Roadex II project. April, 2005 http://www.roadex.org/publications/lowdensity/3_2%20Monitoring%20&%20Focusing%20Tools_l.pdf
13. Tietomekka Ltd. http://www.tietomekka.fi
14. XForms 1.0 (Second Edition) W3C Recommendation 14 March 2006. http://www.w3.org/TR/2006/REC-xforms-20060314/
15. YIT Group. 2006. http://www.yitgroup.com/main.asp?path=1;4620