Modeling Communication Behaviour of Mobile Applications

Volker Gruhn and André Köhler

University of Leipzig, Department of Computer Science,
Chair of Applied Telematics / e-Business, Klostergasse 3, 04109 Leipzig, Germany
{gruhn, koehler}@ebus.informatik.uni-leipzig.de

Abstract. This paper introduces a graphical modeling notation based on coloured petri nets for the performance and cost evaluation of mobile applications. When developing such an application some restrictions due to the low bandwith of mobile networks need to be considered. The notation can be used to model the workflow of an mobile application with just a small effort. The resulting model can be (automatically) transformed into a coloured petri net for simulating the communication behaviour depending on typical user interactions. The simulation results are an important basis for improving the applications performance and the occuring costs for using mobile networks.

Keywords. Modeling languages, Workflow modeling, Mobile and ubiquitous information systems, Web information systems

1 Introduction

Since the availability of mobile broadband networks and the reduced costs for mobile devices the use of mobile applications has become an interesting opportunity in several fields. Companies with large divisions of mobile employees (e.g. service technicians, sales representatives, healthcare services) can use mobile applications to gain access to corporate applications and databases at the point of service (POS). Therewith better coordination of mobile employees, rapid task assignment, the avoidance of error-prone format conversion, instant access to customer data and many more becomes feasible [1], [2]. However, until now just a few companies started using these opportunities. The reason for this chary manner is the characteristics of todays mobile networks (in the following used synonym for radio networks, cellular networks, and wireless networks). Unfortunately they hold some unfavourable qualities which are of particular relevance when trying to enable mobile access to a corporate application:

1. **Long response times.** Compared to the familiar LAN environment mobile networks offer just a very small bandwidth. The large data throughput of a typical corporate application often causes long response times under these bandwidth restrictions. Additionally, in mobile networks the upstream bandwidth is often significantly lower than the downstream bandwidth. That

works for most consumer applications but causes the break down of corporate applications with frequent push operations (e.g. sending documents generated at POS).

2. **High costs.** For the use of a mobile network the provider charges a fee on the basis of the time spent online or the data volume transmitted. The todays provider rates are very expensive and often prevent users especially in the consumer area from using mobile applications. Flat rates for mobile data transmission are available since a short time, but often solve the cost problem just for large companies as they can negotiate reasonable prices due the offering of high user numbers and long-term contracts.
3. **Unstable accessibility.** Within wireless networks all the stability assumptions of the LAN environment are vanishing [3]. It can never be anticipated, if a certain mobile network will be available at a specific place at a specific time. Thus, the time slot for mobile network access as well as the available quality of service (e.g. signal strength, bandwith, costs) are unstable. We assume that in the next five years the number of different mobile networks available at a certain location will grow rapidly especially in urban areas. The increasing number of different mobile network types (e.g. GPRS, EDGE, UMTS, WLAN, WiMAX etc.) combined with some providers can easily lead to the availability of a large number of mobile networks in a certain location, distinguishable in their quality of service.

For a succesful use of a mobile application the circumvention of these shortcomings is crucial. They are caused by the fact that a large amount of data needs to be transmitted via a small bandwidth network. Thus, an application designated for mobile use should be evaluated under consideration of the data volume transmitted for typical workflows at the POS, e.g. through simulation. On the basis of these results an application tuning for data intensive communication sequences can be performed. Obviously, the reduction of the transmitted data volume directly leads to an improvement of the given shortcomings. Therefore, we propose the following method: First, a model of the mobile applications' client-side is needed describing the mobile workers' workflow (or business process) in its single steps at the level of network communication respectively application dialogs. Second, an analysis of the model is needed regarding response times and costs. Third, the application need to be readjusted in order to meet defined performance and cost target values. In the following such a method is presented in detail.

The paper is organized as follows: Section 2 gives an overview about related work. Section 3 shows detailed requirements for the modeling and the analysis of mobile applications under consideration of the above defined situation. Section 4 introduces a petri-net-based graphical modeling notation for the defined purpose. The theoretical work is applied in a case study presented in section 5. Section 6 gives a short conlusion and shows our planings for further research.

2 Related Work

The changes for the discipline of software engineering when developing systems for mobile environments are discussed in [3]. The authors state that "mobility represents a total meltdown of all stability assumptions [...] associated with distributed computing". A comprehensive overview of software engineering for mobile systems is given, regarding issues like models, algorithms, applications and middleware to solve in the future. Our paper adresses the some of these modeling issues. In [4] and [5] a modeling framework to analyze the performance of applications in a mobile computing environment is presented. UML activity diagrams are used for describing the user interaction with the application as well as the network communication. The paper focuses on different scenarios of physical mobility in order to make code mobility based adaptation policies.

A lot of work is done regarding system architectures and other technical aspects of mobile system. An example for this work is [6], where a three-layer software architecture for distributed and mobile collaboration is presented. [7] presents an approach for the modeling and performance evaluation of mobile multimedia systems using generealized stochastic petri nets. The author focuses on verifying the optimal performance achievable under some QoS constraints in a given setting of design parameters. In [8] an architectural model that identifies the components representing the essential aspects of a mobile agent system is described. The interaction design for mobile information systems is subject of [9]. The authors developed a platform that supports the rapid prototyping of multi-channel, multi-modal, context-aware applications and describe how it was used to develop a tourist information system. In [10] and [11] we showed first results for a cost and performance evaluation of mobile applications. There, we stated that it is quite difficult to obtain meaningful values regarding the usage and the data volume for such applications. For this point, a solution is presented in this paper.

3 Requirements for Modeling and Analysing the Communication Behaviour of Mobile Applications

3.1 Defining Evaluation Objectives

When analyzing the communication behaviour of a mobile application its architecture is of particular relevance. According to [12], mainly four different types can be distinguished. One of them is the always online architecture, in which the mobile application communicates with a central server exclusively. This architecture is typical for web-based systems. In the following we limit our explanations to the always online architecture for the sake of brevity.

In order to decide whether a certain application is capable for mobile use, concrete requirements must be defined, most suitable deduced from the mobile workers needs. Therefore the workflow (term used in the following as equivalent for business process) need to be modeled and analysed regarding the use of the mobile application. Thus, the later results can be availed to evaluate the application regarding workflows which e.g. are business critical, have many recurrences, or are important for customer satisfaction. The model should contain at least all steps in the workflow where the mobile application is used. Since the application uses a mobile network, its communication behaviour is of particular interest for the later analysis. Therefore each step in the workflow model should contain detailed information about the communication behaviour of the application which is mainly the data volume transmitted. For a better understanding of these contiguous aspects a short formal description is given in the following.

Be P a set of network providers and N a set of network types. Each combination $(p, n) \in P \times N$ is called *wireless network*. The volume-based fee $f(p, n)$ for a certain unit for the usage of a network depends on the wireless network $(p, n) \in P \times N$. The maximum upstream bandwidth $b_{up}(p, n)$ and the maximum downstream bandwidth $b_{down}(p, n)$ depend also on the mobile network $(p, n) \in P \times N$. A workflow consists of a set of activities A. One activitiy $a \in A$ is characterized by the average sent data volume $dv_{sen}(a)$ an the average received data volume $dv_{rec}(a)$.

Based on these correlations, certain evaluation objectives can be deduced. For each activitiy $a \in A$ the average request time $t_a^s(p, n)$ as well as the average response time $t_a^r(p, n)$

$$t_a^s(p, n) = \frac{dv_{sen}(a)}{b_{up}(p, n)} \qquad t_a^r(p, n) = \frac{dv_{rec}(a)}{b_{down}(p, n)} \tag{1}$$

can be calculated. The overall transmission time of the activitys' data volume is given through:

$$t_a(p, n) = t_a^s(p, n) + t_a^r(p, n) \tag{2}$$

For an activity $a \in A$ the volume-based fee for the sent data volume can be calculated with $c_a^s(p, n)$ and for the received data volume with $c_a^r(p, n)$:

$$c_a^s(p, n) = dv_{sen}(a) \cdot f(p, n) \qquad c_a^r(p, n) = dv_{rec}(a) \cdot f(p, n) \tag{3}$$

The volume-based fee for the overall transmitted data volume for an activity $a \in A$ is given through:

$$c_a(p, n) = c_a^s(p, n) + c_a^r(p, n) \tag{4}$$

For each workflow activity $a \in A$ the average number of recurrences r_a can be calculated, e.g. through simulation. Thus, the average costs c_w for one workflow recurrence is given by:

$$c_w(p, n) = \sum r_a \cdot c_a(p, n) \text{ for each } a \in A \tag{5}$$

Therewith seven basic criteria for the evaluation of a mobile application are defined. Of course this model could be extended with more aspects, e.g. allowing the change of the used network between activities, distinguishing costs on the activity level, allowing for volume-dependent fees, and modeling asynchronous

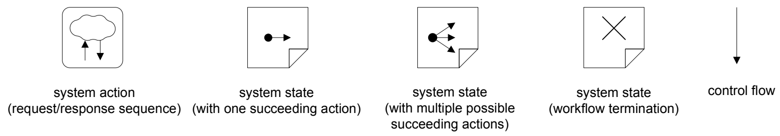

Fig. 1. Notation elements for web-based applications

communication. As stated above, we focus in this paper on web-based applications and therefore consciously introduced just basic evaluation criteria needed for the understanding of the further explanations. In the following, it is shown how to model the communication behaviour of the application in order to obtain meaningful values for the above defined evaluation criteria.

4 Petri-net-based Graphical Modeling Notation

For the modelling of workflows plenty of notations are available, but just a few allow the simulation of the created model in order to obtain data regarding the workflow and the applications' communication behaviour. Considering this, we decided to use coloured petri nets (CPN), which have a complete formal basis, allow a very flexible modelling, and tools for the modelling and the simulation are available [13], [14]. The advantage of the great flexibility and the excellent expressiveness unfortunately causes also the biggest disadvantages of CPN for our purpose. First, even for small applications the model grows fast and becomes very complex and almost unmanageable. Second, the creation of the model requires extensive knowledge and practice regarding the modeling of a CPN. We do not assume, that the user of our method will have these qualities. Thus, we developed a simple notation that covers the complexity of CPN for our particular purpose. The elements of this notation, needed for modeling web-based applications, are introduced in the following. For the sake of brevity, we give no formal introduction to CPN (see rather [13]), but later CPN examles are explained informal so that also readers having no CPN knowledge will understand their basic functionality.

4.1 Graphical Notation Elements: Overview

The notation elements consist of system actions, systems states, and control flow (see Fig. 1). A system action is illustrated by a square with rounded corners describing one request/response sequence via a mobile network, between client and server. A system state is illustrated by a square document symbol as it mostly represents an application dialog. It describes the system state between two system actions where no communication via the mobile network occurs and the mobile worker interacts with the application. The control flow connects system actions and system states, and shows their execution order. A system actions' predecessor(s) as well as the successor(s) must be one or more system states. Equally, a system states' predecessor(s) as well as the successor(s) must be one or more system actions.

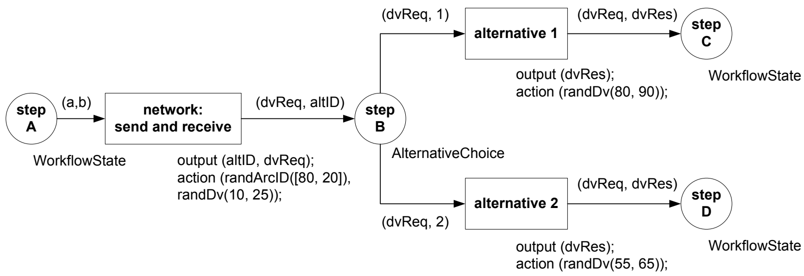

Fig. 2. CPN defining the system action (request/response cycle)

4.2 System Action (request/response sequence)

For modeling web-based applications just one system action is needed, as communication via the mobile network occurs always as a request/response sequence. The corresponding graphical element contains a cloud representing the network to communicate with (see Fig. 1, utmost left hand side). The starting point of this system action is a system state from which the mobile worker initiates a network request, e.g. through clicking a hyperlink or submitting formular data. In case of clicking a hyperlink, the following response is usually clearly defined and leads to one possible system state (e.g. new webpage). However, in dynamic applications the result may vary over a certain range of defined alternatives depending on the user input. The system action simulates the incidence of one alternative with given probabilities. Furthermore, for both the request and the response it calculates the transmitted data volume from a given range.

This action is defined by a place-bounded CPN as shown in Fig. 2. The used data types and variables in this CPN are defined as follows:

```
colset DataVolume = int;
colset DataVolumeReq = int;
colset DataVolumeRes = int;
colset AlternativeID = int;
colset AlternativeChoice = product DataVolumeReq * AlternativeID;
colset WorkflowState = product DataVolumeReq * DataVolumeRes;

var altID : AlternativeID;
var a, b, dvReq, dvRes : DataVolume;
```

A token in this CPN represents the actual position in the workflow and carries information regarding the transmitted data volume of the current communication sequence. If a token from type *WorkflowState* is put into place *step A* the transition *network: send and receive* fires. In doing so, the function *randArcID* randomizes from the given alternatives. For each alternative the probability is assigned, the overall sum must be equal to 100. The example in Fig. 2 consists of one alternative with probability 0.8 and a second alternative with probability 0.2. Furthermore, the function *randDv* randomizes the sent data volume $dv_{sen}(a)$ from a given range. Finally, the transition creates a token as output for place

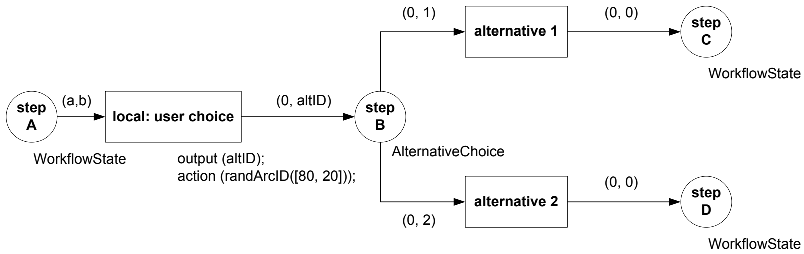

Fig. 3. CPN defining the system state (with multiple succeeding system actions)

step B carrying the result of the function *randArcID* as *altID* and the result of function *randDv* as *dvReq*. The place *step B* has one or more succeeding transitions depending on the defined alternatives. As each alternative has an ID, the token from *place B* can just be processed by the transition that matches the tokens' *altID*. The corresponding condition is specified at the outgoing arc of *place B*. The transition *alternative x* randomizes the received data volume $dv_{rec}(a)$ from a given range and stores it with the token as *dvRes*. The token now carries information about the sent data volume *dvReq* and the received data volume *dvRes* and is put into the following place (resulting system state). In Fig. 2 two alternatives are shown, but any number of alternatives is feasible. As we aim on hiding the complexity of CPN, the user can apply this net through the use of the appropriate symbol and need to specify just the variable attributes, which are the actions' name, a range for the requested data volume, the number of alternatives, a probability for each alternative, and a range for the received data volume for each alternative.

4.3 System State (with multiple possible succeeding actions)

In most of the cases, a system state will have multiple possible succeeding actions, e.g. a couple of hyperlinks. In this case each succeeding action will be choosen by the mobile worker with a certain probability. Hence, this system state is defined by a CPN as shown in Fig. 3. This system state is illustrated by a document symbol showing some arrows (see Fig. 1, third from left hand side). The used data types and variables in this CPN are the same as the ones above. The functionality of this CPN is slightly similar as the one explained before. The starting point is a system state, represented by a place (in Fig. 3 *step A*). If a token from type *WorkflowState* is put into this place the transition *local: user choice* fires. It randomizes from the given alternatives and puts the resulting *altID* on the token, placing it into *place B*. From there, only the corresponding transition having the right alternative ID at its incoming arc is able to fire and puts a token in the following place describing the choosen alternative. From there, the corresponding system action could start immediately. In Fig. 3 two alternatives are shown, but any number of alternatives is feasible. As we aim on hiding the complexity of CPN, the user can apply this net through the use of the appropriate symbol and need to specify just the variable attributes, which

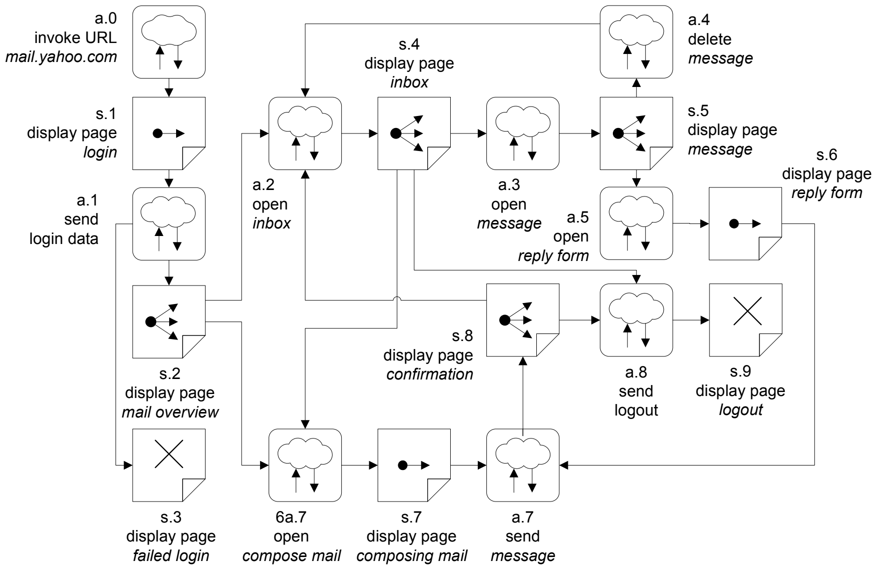

Fig. 4. Yahoo! mail portal

are the states' name, the number of alternatives, and a probability for each alternative.

4.4 System State (with one succeeding action and workflow termination)

A system state having one possible succeeding system action (e.g. one single hyperlink) does not need to be defined by a CPN. In fact, it matches with one of the source places respectively the drain places from a system action. Such a state is illustrated by a document symbol showing a single arrow (see Fig. 1, second from left hand side).

Each workflow has a defined end which is in our case specified through a system state. This state is illustrated by a document symbol showing a cross (see Fig. 1, second from right hand side). This state does not need to be defined by a CPN as it matches with one of the drain places from the preceding system action.

5 Case Study

In the following we show, how the above introduced modeling method was applied for the yahoo mail portal. Many people are interested in accessing their mails also when they are away, e.g. using a notebook and GPRS or UMTS adapters. Thus, we would like to answer the question whether such mobile networks have sufficient response times for the given application and what costs

ID	name	request dv_{sen} in kb		response alt. 1 dv_{rec} in kb			response alt. 2 dv_{rec} in kb		
		min	max	prob	min	max	prob	min	max
a.0	invoke URL yahoo.com	10	20	1.0	60	80	0.0		
a.1	send login data	40	60	0.9	80	105	0.1	57	95
a.2	open inbox	10	20	1.0	30	120	0.0		
a.3	open message	10	20	1.0	180	260	0.0		
a.4	delete message	10	20	1.0	80	160	0.0		
a.5	open reply form	10	20	1.0	180	260	0.0		
a.6	open compose mail	10	20	1.0	185	185	0.0		
a.7	send message	10	20	1.0	90	200	0.0		
a.8	send logout	10	20	1.0	30	50	0.0		

Table 1. Specification of system actions

		probability of alt. 1	probability of alt. 2
s.1	display page login	1.0	0.0
s.2	display page mail overview	0.9	0.1
s.3	display page failed login	0.0	0.0
s.4	display page inbox	1.0	0.0
s.5	display page message	0.2	0.8
s.6	display page reply form	1.0	0.0
s.7	display page composing mail	1.0	0.0
s.8	display page confirmation	0.8	0.2
s.9	display page logout	0.0	0.0

Table 2. Specification of system states

ID	system action	recurrences	GPRS $t \leq 2s$	GPRS $2s < t \leq 5s$	GPRS $t > 5s$	UMTS $t \leq 2s$	UMTS $2s < t \leq 5s$	UMTS $t > 5s$	WLAN $t \leq 2s$	WLAN $2s < t \leq 5s$	WLAN $t > 5s$
a.0	invoke URL yahoo.com	50	0	3	47	50	0	0	50	0	0
a.1	send login data	50	0	50	0	50	0	0	50	0	0
a.2	open inbox	74	0	0	74	74	0	0	74	0	0
a.3	open message	33	0	0	33	2	31	0	33	0	0
a.4	delete message	3	0	0	3	3	0	0	3	0	0
a.5	open reply form	30	0	0	30	4	26	0	30	0	0
a.6	open compose mail	35	0	0	35	20	15	0	35	0	0
a.7	send message	30	0	0	30	12	18	0	30	0	0
a.8	send logout	48	0	48	0	48	0	0	48	0	0

Table 3. Simulation results

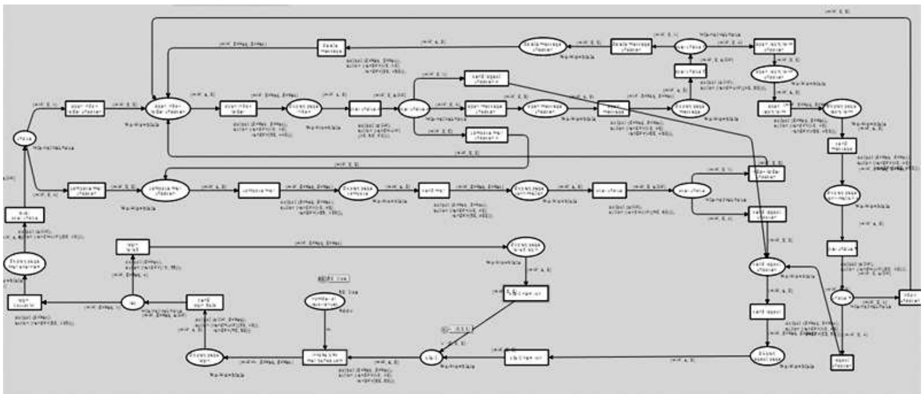

Fig. 5. Resulting petri net

occur when using them. First, we determined typical workflows that are of particular interest for us. That is *checking the inbox for new messages*, *open a message*, *delete a message*, *reply to a message*, and *compose a message*. Then we created a model of the mail portal using the above introduced notation. The result is shown in Fig. 4. The workflow starts with the invocation of the URL (network request/response cycle a.0). The server response contains the data for displaying the login page in which the user enters the login data (system state s.1). By clicking the submit button, the login data is sent to the server (a.1). Two server responses are possible: First, the login failed and the client shows a suitable page (s.3). Second, the login is successful and a mail overview page is displayed (s.2). From this point, several user actions are possible. If the user chooses to compose a new mail (a.6), the appropriate page is displayed (s.7). By clicking the send button, the message is sent (a.7) and a confirmation page is displayed (s.8). From there, the user can choose between returning to the inbox folder or to log out. In the latter case (a.8), a confirmation page is displayed (s.9). From the inbox folder (s.4) a message can be opened (a.3) and gets displayed (s.5). This message can be deleted (a.4), the user then returns to the inbox folder (a.2). The user can also reply to this message (a.5) using a reply form (s.6). The further steps are the same as described in *composing message*.

After creating this model, we need to specifiy arc probabilities wherever different alternatives occur and we need to define minimum and maximum values for the expected data volume which is sent and received. We collected the latter data with a HTTP protocol sniffer tool. The resulting data is showed in Table 1 (values for system actions) and in Table 2 (values for system states). The created model can now be transformed into the corresponding petri net according to the above defined subnets. The resulting net is shown in Fig. 5. This petri net can now be used for a simulation of a large number of workflow recurrences. We did such a simulation with CPN-Tools [14] and then analyzed the simulation results regarding the evaluation objectives defined in section 3.1. The results are shown in Table 3. We conducted the calculation assuming an average up- and downstream bandwidth for three common mobile network types, i.e. GPRS (15 kb/s), UMTS (100 kb/s), and WLAN (500 kb/s). Furthermore, we defined a target response time of two seconds and a maximum response time of five seconds. If a communication sequence' response time lasts up to the threshold value of two seconds, we rated this application piece as mobile usable. If the response time lasts up to five seconds, we rated this application piece as generally mobile usable but with needs for optimization. If the response time lasts longer than five seconds, we rated this application piece as not mobile usable. The simulation of the net was done 50 times represented by 50 tokens in the CPN. The results show, that considering the above described restrictions the yahoo mail portal is not usable on mobile devices using a GPRS network connection. Using an UMTS network connection, the mail portal works fine in most of the cases. For the actions *open message*, *open reply form*, *open compose mail*, and *send message* some optimization potential was found. In the WLAN environment, the application performs fine for each system action, as this is the environment the application is originally build for. For the sake of brevity we abstain from showing the results of the cost evaluation. Considering these results, the application can now be modified, if a use for e.g. UMTS is intended. The results show, that especially the requests for opening a message as well as for opening the reply form would perform badly in most of the cases. The application tuning should start at these points.

6 Conclusion and Further Research

When developing applications for their use with mobile networks, bandwith limitations and occuring costs need to be considered in order to assure the usability of the application at the POS. For already existing applications originally designed for usign in the LAN environment one need to cope with the same challenges when trying to make these applications available at the POS. Therefore, we presented a lightweight graphical modeling notation that can be used for modeling workflows within mobile applications. This notation is based on coloured petri nets and allows the simulation of the created model regarding the frequency of communication sequences as well as their costs and performance. The results of the simulation can be used for aligning the application to business and usability requirements through tuning the application at the right points.

Further research is planned in several fields regarding the shown method. First, we plan to extend the underlying evaluation model, e.g. through considering different bandwiths for up- an downstreams, allowing asynchronous communication etc. Second, we want to extend also our modeling notation in order to be able to describe more mobile application types, e.g. asynchronous communication with rich clients and offline clients with occasionally synchronization. Third, we work on a tool support for our modeling notation that will allow the automatically transformation of the created model into a CPN.

References

1. Gruhn, V., Köhler, A., Klawes, R.: Modeling and analysis of mobile service processes by example of the housing industry. In van der Aalst, W.M., Benatallah, B., Casati, F., Curbera, F., eds.: Business Process Management, Springer LNCS 3649 (2005) 1–16
2. Nah, F.F.H., Siau, K., Sheng, H.: The value of mobile applications: a utility company study. Volume 48., New York, NY, USA, ACM Press (2005) 85–90
3. Roman, G.C., Picco, G.P., Murphy, A.L.: Software engineering for mobility: a roadmap. In: ICSE '00: Proceedings of the Conference on The Future of Software Engineering, New York, NY, USA, ACM Press (2000) 241–258
4. Grassi, V., Mirandola, R., Sabetta, A.: Uml based modeling and performance analysis of mobile systems. In: MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems, New York, NY, USA, ACM Press (2004) 95–104
5. Grassi, V., Mirandola, R.: Primamob-uml: a methodology for performance analysis of mobile software architectures. In: WOSP '02: Proceedings of the 3rd international workshop on Software and performance, New York, NY, USA, ACM Press (2002) 262–274
6. Dustdar, S., Gall, H.: Architectural concerns in distributed and mobile collaborative systems. Volume 49., New York, NY, USA, Elsevier North-Holland, Inc. (2003) 457–473
7. Tsang, T.: Modelling and performance evaluation of mobile multimedia systems using qos-gspn. Wirel. Netw. **9** (2003) 575–584
8. Schoeman, M., Cloete, E.: Architectural components for the efficient design of mobile agent systems. In: SAICSIT '03: Proceedings of the 2003 annual research conference of the SAICSIT, Republic of South Africa, South African Institute for Computer Scientists and Information Technologists (2003) 48–58
9. Belotti, R., Decurtins, C., Norrie, M.C., Signer, B., Vukelja, L.: Experimental platform for mobile information systems. In: MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking, New York, NY, USA, ACM Press (2005) 258–269
10. Book, M., Gruhn, V., Hülder, M., Köhler, A., Kriegel, A.: Cost and response time simulation for web-based applications on mobile channels. In: QSIC '05: Proceedings of the Fifth International Conference on Quality Software (QSIC'05), Washington, DC, USA, IEEE Computer Society (2005) 83–90
11. Book, M., Gruhn, V., Hülder, M., Köhler, A.: Performance tuning and cost discovery of mobile web-based applications. In Mendes, E., ed.: Proceedings of the ICWE 2005 Workshop on Web Metrics and Measurement, University of Wollongong, School of IT and Computer Science (2005) 33–37
12. Book, M., Gruhn, V., Hülder, M., Schäfer, C.: A methodology for deriving the architectural implications of different degrees of mobility in information systems. In H. Fujita, M., ed.: New Trends in Software Methodologies, Tools and Techniques, IOS Press (2005) 281–292
13. Jensen, K.: Coloured Petri nets (2nd ed.): basic concepts, analysis methods and practical use: volume 1. Springer-Verlag, London, UK (1996)
14. (http://wiki.daimi.au.dk/cpntools/cpntools.wiki)