

Predicting the Usability of Telemedicine Systems in Different Deployments through Modelling and Simulation

Cristina-Adriana Alexandru and Perdita Stevens

School of Informatics
University of Edinburgh

Abstract. The success of telemedicine systems requires sufficiently high productivity, satisfaction and low error rates: in short, usability. However, the costs associated with performing pre-deployment usability studies often make them infeasible. This results in usability problems which may manifest differently in different deployment contexts and cause different levels of inefficiency and discomfort to their users, potentially even leading to errors which may threaten patients' lives. This paper shows by means of examples how experimentation with different workloads, user profiles and system characteristics can suggest problems that may arise in deploying a telemedicine system into new contexts. Such approaches might support the roll-out of large telemedicine implementations, as the one currently planned in the UK.

Keywords: telemedicine, telemonitoring, usability, efficiency, scaling up, cognitive modelling, simulation

1 Introduction

By better considering the user's profile, work context and needs, good usability in a computer system makes a user's interaction more fit for purpose and intuitive [7]. This leads to a quicker learning of the system's functionality with less need for training and support, increased productivity, reduced error rates and increased satisfaction. All of these aspects are important for today's telemedicine systems, intended as a solution to be adopted at scale for the management of patients suffering from long-term conditions in an ageing population [1]. In particular, while reduced error rates are essential for any safety-critical system, increased productivity through the more timely high-quality remote care of the increasing patient population is one of the main purposes for which telemedicine

Copyright ©2013 by the paper's authors. Copying permitted for private and academic purposes.

In: H. Gilstad, L. Melby, M. G. Jaatun (eds.): Proceedings of the European Workshop on Practical Aspects of Health Informatics (PAHI 2013), Edinburgh, Scotland, UK, 11-MAR-2013, published at <http://ceur-ws.org>

systems are being introduced. Moreover, such systems must be acceptable to their users (nurses, general practitioners, clinical or non-clinical monitors) before telemedicine could be proven to be the way forward. We strongly believe that some of the organisational changes brought about by telemedicine (e.g. in what concerns resources used or the daily work practices) can be partly addressed by a better consideration of usability in telemedicine system design [4]. Moreover, improved usability can reduce potential user reticence towards the used system and encourage acceptance.

Despite the recognised importance of usability, it is often uneconomical to perform pre-deployment usability studies of telemedicine systems, especially if these systems are to be deployed in several contexts which would require an evaluation in each. This results in usability problems which can cause frustration, inefficiency or lead to errors which may threaten patients' lives. Such usability problems may be hard to foresee, especially if the system is to be deployed in several different contexts. As observed by Alexandru in a post-deployment usability evaluation of two web-based telemonitoring systems used as part of medical trials in volunteering practices ([3]), usability problems may manifest differently in each deployment context of the system: large variations in the effective usability of the systems were observed between and even within healthcare centres, due to such factors as differences in the number of patients being handled by each user; the characteristics of the user and the user's pattern of work (e.g. a doctor using the system occasionally versus a practice nurse using it regularly); the characteristics of the monitored patient group; the quality of internet connection available, etc.

The UK is planning a large-scale telemedicine implementation as part of the Delivering Assisted Living Lifestyles at Scale (DALLAS) programme ([2]). Similar developments are occurring in other countries. This being the case, a cost-effective alternative to the often avoided pre-deployment usability evaluation of a telemedicine system becomes desirable for helping to predict the outcomes of the system in each deployment context such that resources are not wasted.

This paper builds on previous work ([4]). There, we proposed a methodology, together with a modelling approach, for helping predict, based on the detected usability problems from a reference deployment site and knowledge about a potential deployment site, whether and which usability problems would appear in the potential deployment site. In this paper we briefly outline our approach and provide examples highlighting its advantages. These examples focus on the efficiency component of usability for a telemonitoring website and are based on experience from previous work ([3]). They contain invented, but realistic facts and numbers. The approach will be thoroughly evaluated by assessing a real telemonitoring system in use in the near future.

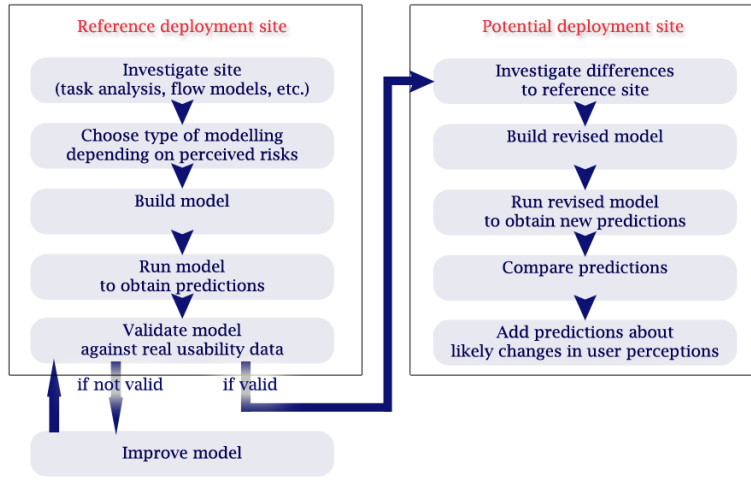


Fig. 1. The methodology[4]

2 Brief overview of the approach

2.1 The methodology

Our methodology is intended as a guide for modellers for extending the scope of usability evaluation from a reference deployment site to making predictions about usability in potential deployment sites based on information about these new contexts only. This offers the great advantage of reduced costs as opposed to needing to perform a new usability evaluation in each site, helps to understand the level of risk associated to each context for deciding whether it is worth investing in it, and helps to identify and prioritise those areas of the system or process which need changing. The methodology may be used with any type of modelling approach depending on the type of perceived risks to the system.

By starting from a detailed analysis of the user's profile, needs, work environment and usability problems of the system within a reference deployment site, we decide on the major perceived risks and choose an appropriate modelling approach. We build the model, instantiate it with input data collected from the context and run it, then compare its predictions with identified usability issues. It is then improved and rerun until its predictions are deemed good enough. Whenever the use of the system in a potential deployment site is considered, we investigate the user and environment differences between the reference context and this new context and reconstitute our validated model with the acquired data. A new run will now provide an indication of the likely differences in the perceived usability of the two sites. A step-by-step representation of the methodology is provided in Fig. 1.

2.2 The modelling approach

To illustrate our methodology in action, we have developed a system that can run a model as described above. To "run a model" means to use models of the user of a telemedicine system and of the behaviour of the system itself, together with statistical information about key aspects of the environment, to generate predictions about what will happen when the system is used in practice. This is done by repeatedly simulating the user's interaction with the system, each time with a different set of input data representing the variable facts about the world that may affect the user and system, e.g., what the blood pressure readings of the patients monitored are. By systematically generating these input data, running the simulations, and collecting results, we can answer questions about the system's efficiency (e.g. How long does it usually take for the user to manage her workload? Can she usually manage her workload in her available time?).

The modeller must describe:

1. User workload and other relevant aspects of the environment, from which input data for a given simulation will be generated. These will be numeric, and may be constants (the same on every simulation run, e.g. the number of patients to be monitored) or parameters for distributions (from which numbers for each simulation run will be drawn, e.g. blood pressure readings).
2. The user's profile in terms of goals when using the system, knowledge (e.g. as related to a monitoring task) and skills (e.g. how to monitor a patient).
3. The system in terms of what it displays to the user or keeps as internal at one time and how this changes once a user performs an action on it
4. Time-related data: total time available to the user, specific time she spends for performing actions internal or external to the system, system response time for each action taken on it

The inputs from point 2 and specific time-related user data from point 4 from above are used by our modelling approach to instantiate a user model which can simulate the user's goal-driven behaviour on the system. This model is inspired from the Icarus cognitive architecture ([15, 6, 12, 13, 14]). The inputs from point 3 and specific time-related system data from point 4 are used to instantiate a system model, modelled as a basic labelled transition system. The two models work in parallel during a run of the approach.

Each run takes a new workload generated from the inputs from point 1 by directly using the given numbers or drawing from the distributions as appropriate. A sufficiently high number of repeated runs can provide evidence as to the expected time distribution for user work within the deployment context being modelled. This result can be compared with the user's total available time from the inputs in point 4 such that the case of the user exceeding this time is flagged up. We can change the models and inputs to explore different deployment contexts (in terms of workload, user profile and way of doing things) or changes to the system. By performing repeated runs in each case, we can obtain comparative predictions about efficiency, as we exemplify below.

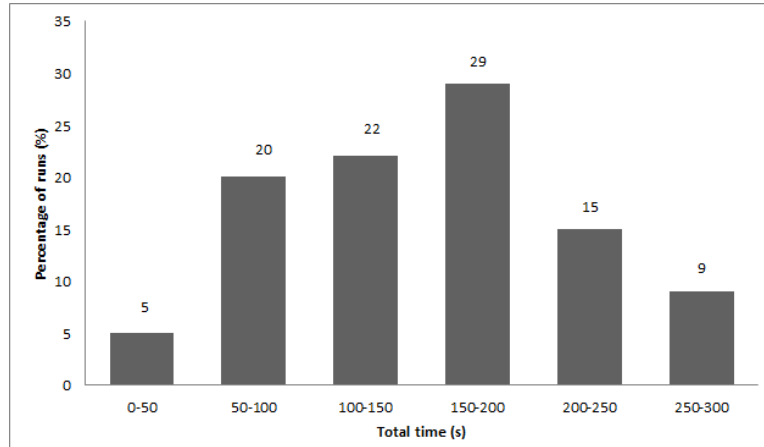


Fig. 2. Total time to complete workload for scenario 1

3 Examples

To demonstrate the usefulness of our approach, let us consider the practitioner interface to an online telemonitoring system which is weekly used by nurses for monitoring patients suffering from hypertension. Let us suppose that, according to an initial analysis of a reference deployment site (e.g. practice or hospital), we find that the system is used for routinely monitoring 10 patients whose reading criticality is characterised by a mean of 30% and standard deviation of 10%. We consider the following **first scenario**:

The nurse selects patients who are flagged up as potentially needing intervention from an unordered table on the homepage by hovering the mouse over them and clicking on 'Select' from a pop-up box; this takes her to the patient details page, where she can see the patient's demographics, last two days of readings (represented as systolic-diastolic pairs) and the any taken notes. If the patient's last two days of readings are only closely exceeding limits, no action is needed and the nurse returns to the homepage by clicking on an appropriate button. If the patient's last two days of readings are more critical, the nurse clicks on a button to be taken to the patient's last five days of readings. If most of these readings are exacerbations, the nurse needs to go back to the patient details page by clicking on an appropriate button, enter a note on the page about her assessment to be later used for setting up an appointment for the patient, and return to the homepage. If however few of the readings on the detailed readings page are exacerbations, no action is needed and the nurse returns to the homepage by clicking on a button. The nurse writes down each patient's name on paper to remember she was checked before continuing with the next patient in the table. Given the nurse's screen resolution, 10 patients from the homepage are visible at one time without the need to scroll, therefore all the patients for this scenario.

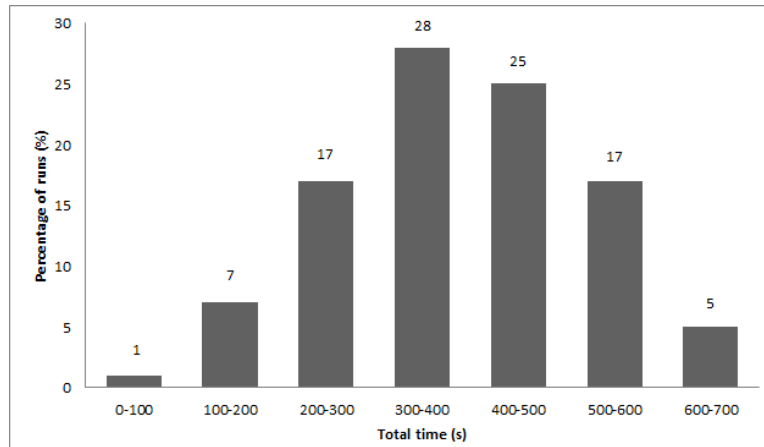


Fig. 3. Total time to complete workload for scenario 2

Having appropriately instantiated the user and system models we run 100 simulations to get statistics about the total time it takes for the nurse to monitor all of the patients. The outcome is presented in Fig 2. The graph shows that in almost half of the runs (47%) the nurse will accomplish her monitoring work within 150 seconds and that for more than three quarters of the runs (76%) she will accomplish it within 200 seconds.

Let us next consider that there are plans to deploy the same system within a context which differs in the number of patients: 20 instead of 10, but otherwise is similar to the reference deployment site (**second scenario**). Although one would normally assume that double the patients would mean double the time spent on average on monitoring them (therefore 50% of runs within around 300 seconds), by running another 100 simulations with the changed input we obtain the unexpected graph from Fig. 3.

This result clearly shows that the real average run time lies close to 400 seconds (in 53% of the runs the work will take less than 400 seconds) and that in 28% of the runs the work will take between 300 and 400 seconds, while in only 25% of the runs it will take less than 300 seconds. This is due to the time spent by the nurse scrolling down to find each of the patients who needs an intervention and is not visible in the initial position of the homepage- all the patients from the 11th to the 20th.

The table on the homepage being unordered, the nurse will normally need to scroll over many unflagged patients to find the patients she needs to check. To improve on this, let us assume that the software company adds an ordering option which places flagged up patients (those needing an intervention) at the top of the table. This option will need to be used only once by the user within a work session and then will be saved for the session. For monitoring all of the flagged up patients, nurses will now need to sort the table and only monitor those patients who are flagged up, having finished their work once they get to

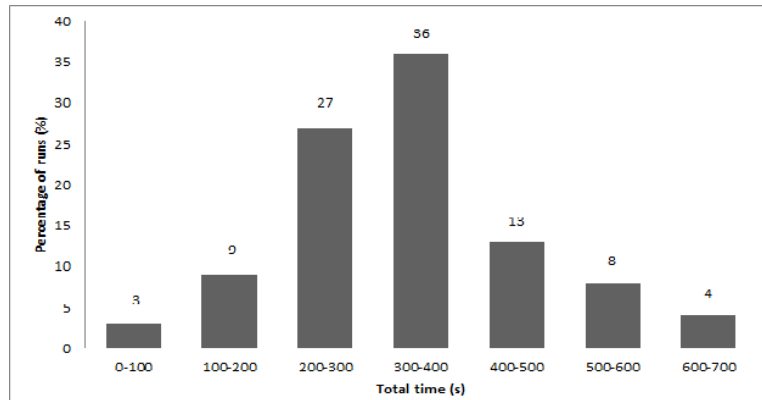


Fig. 4. Total time to complete workload for scenario 3

the first unflagged patient. By changing the description of the behaviour of the system and the user skills to include the sorting action and rerunning another 100 simulations with this **third scenario**, where we have kept the other inputs as in scenario 2, we obtain the graph from Fig. 4.

The figure demonstrates an improvement as compared to Fig. 3, as we can see that in 75% of the runs the nurse's work will take less than 400 seconds, while this happened only in 53% of the runs in the case of the unordered table of the second scenario. Also, in 39% of the runs the work takes less than 300 seconds as compared to 25% of the runs in the second scenario. This makes the average time to accomplish the work be much lower than in the previous scenario and much closer to the 300 seconds expected (double the time of the first scenario). We have therefore demonstrated that the addition of a simple ordering function helps save time. It would also clearly reduce the frustration caused by the need to scroll to find all of the flagged up patients from the previous scenario.

Although these simple examples allow only for differences in terms of minutes to be observed, it proves the usefulness of our approach for showing how contexts characterised by different workloads and the usability of a telemedicine system given by different design decisions can influence the time spent by a user in her work.

4 Related Work

Surprisingly we have not found work proposing a methodology similar to ours in this field or others. Areas where this might be expected include performance modelling and business process change management. The closest, specific to performance modelling, is [5]. A strand of work combining the two fields includes such papers as [8, 9] by authors at SAP, and investigates how to answer what-if questions concerning specific changes to business process models; however, this work concerns individual changes to one deployment site rather than comparing

deployment sites that may differ in multiple respects, and is much more specific in scope than our proposal.

Since the high cost of usability evaluation has long been a recognised problem, there has been a considerable amount of work on automating usability evaluation. A survey which is now over ten years old but still seems to cover the main categories of work is [10]. Although user models are widely used to simulate use of computer systems including web applications, the user model is generally fixed while the system, or system model, varies to show how different designs influence the system's usability. In contrast, our modelling approach allows both of the models to vary by being given different inputs reflecting different deployment contexts to explore how the user profile, workload and system design can influence usability.

In what concerns the efficiency component of usability, there is of course a whole literature on using models for predicting a system's or process's efficiency in terms of execution time, the GOMS family of models being the most widely cited in this respect [11]). We are building on this work, the novelty of our modelling approach being its application to make comparative predictions in different, scaling up contexts, which is especially relevant for telemedicine due to reasons of cost and user acceptance.

5 Conclusion and Future Work

We have described a methodology and associated modelling approach which can be used for predicting the usability of a telemedicine system in different contexts. We have shown how instantiating models of a user and system for a reference context and running a high number of simulations, each with different workload characteristics, can help predict the system's efficiency in that context. This can then be repeated with different inputs for the models, representing changes in the context (different workloads, user profiles or way of doing things) or in the system's design, to help us explore differences in efficiency. We have demonstrated the usefulness of our approach by means of examples.

The next step is to evaluate our approach by assessing a real telemedicine system in use, which will allow us to validate our work and incorporate improvements both into its logic and the ease of specification of models. It would be interesting in the light of evaluation to see what the pros and cons of our approach are. We will also use our approach for predicting the efficiency of the wider work process, and not only that of the system- e.g. to predict whether current healthcare staff would still manage their monitoring work in their available time, or more staff would need to be employed, during an epidemic.

References

- [1] Telemedicine for the benefit of patients, healthcare systems and society. European commission staff working paper, European Commission, 2009. Available from http://ec.europa.eu/information_society/activities/health/docs/policy/telemedicine/telemedecine-swp_sec-2009-943.pdf.

- [2] Dallas- delivering assisted living lifestyles at scale. Technical report, Technology Strategy Board, June 2012. Available from http://www.innovateuk.org/_assets/0511/sbri_comp_dallas.pdf.
- [3] C.A. Alexandru. Usability testing and improvement of telemedicine websites. Master's thesis, University of Edinburgh, School of Informatics, 2010. Available from <http://www.inf.ed.ac.uk/publications/thesis/online/IM100870.pdf>.
- [4] C.A. Alexandru and P. Stevens. Models to extend the scope of usability testing for telemedicine systems. In *Digital Futures 2012: The Third Annual Digital Economy All Hands Conference*, Aberdeen, October 2012. Available from http://www.de2012.org/sites/default/files/digitalfutures2012papers/Papers/Session4CUsability/Alexandru&Stevens_TelemedicineSystems.pdf.
- [5] D. H. Bailey and A. Snavely. Performance modeling: Understanding the past and predicting the future. In *Euro-Par 2005 Parallel Processing Lecture Notes in Computer Science*, volume 3468, pages 185–195, 2005.
- [6] D. Choi, M. Kaufman, P. Langley, N. Nejati, and D. Shapiro. An architecture for persistent reactive behavior. *International Joint Conference on Autonomous Agents and Multiagent Systems*, 2:988–995, 2004.
- [7] X. Faulkner. *Usability Engineering*. MacMillan, illustrated edition, 2000.
- [8] M. Fritzsche and J. Johannes. Models in software engineering. chapter Putting Performance Engineering into Model-Driven Engineering: Model-Driven Performance Engineering, pages 164–175. Springer-Verlag, Berlin, Heidelberg, 2008.
- [9] M. Fritzsche, M. Picht, W. Gilani, I. Spence, J. Brown, and P. Kilpatrick. Extending bpm environments of your choice with performance related decision support. In *Proceedings of the 7th International Conference on Business Process Management, BPM '09*, pages 97–112, Berlin, Heidelberg, 2009. Springer-Verlag.
- [10] M. Y. Ivory and M. A. Hearst. The state of the art in automating usability evaluation of user interfaces. *ACM Comput. Surv.*, 33(4):470–516, December 2001.
- [11] B. E. John and D. E. Kieras. The goms family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction*, 3:320–351, 1996.
- [12] P. Langley and D. Choi. A unified cognitive architecture for physical agents. In *Proceedings of the 21st national conference on Artificial intelligence - Volume 2, AAAI'06*, pages 1469–1474. AAAI Press, 2006.
- [13] P. Langley, D. Choi, and S. Rogers. Acquisition of hierarchical reactive skills in a unified cognitive architecture. *Cognitive Systems Research*, 10(4):316–332, 2009.
- [14] P. Langley, D. Choi, and N. Trivedi. Icarus user's manual, July 2011. Available from http://www.dongkyu.com/icarus-tutorial/ICARUS_manual_3.pdf.
- [15] P. Langley, K. Cummings, and D. Shapiro. Hierarchical skills and cognitive architectures. In *Proceedings of the Twenty-Sixth Annual Conference of the Cognitive Science Society*, pages 779–784, 2004.