
An In Situ Data Collection Platform for Personal Informatics

Nikolaos Batalas
Eindhoven University of
Technology
Den Dolech 2, 5600MB
Eindhoven, The Netherlands
n.batalas@tue.nl

Panos Markopoulos
Eindhoven University of
Technology
Den Dolech 2, 5600MB
Eindhoven, The Netherlands
p.markopoulos@tue.nl

Abstract

In this workshop paper we support the position for a more holistic view of experience sampling and personal informatics software. We also discuss the use of our platform, Tempest, as a means for prototyping Personal Informatics applications.

Author Keywords

Experience Sampling, Personal Informatics

ACM Classification Keywords

H.5.m [INFORMATION INTERFACES AND PRESENTATION (e.g., HCI)]: Miscellaneous.; D.2.1 [Requirements/Specifications (D.3.1)]: Elicitation methods (e.g., rapid prototyping, interviews, JAD)

Introduction

Applications that aim to collect data as user go about their daily lives, have long been employed in the service of diary studies or Experience Sampling Protocols, to help researchers derive meaning as to what their participants do or how they feel. Not too long ago, their design space has expanded beyond the confines of research purposes. Incarnated as personal informatics applications, in situ data collection software can be put to the task of helping individuals research their own lives, by monitoring their habits, thoughts and activities and gain self-knowledge

about a very varied array of aspects (e.g. fitness, entertainment, sociability, etc).

Applications for data collection both for personal and for research purposes, have first and foremost in common that they need to collect information from and about the user, in situ, over long periods of time, either by having the user provide this data explicitly (e.g. through a graphical interface) or collect it implicitly (e.g. logging sensor data). As such they share many functional and development requirements. Indeed, the position that ESM tools (coming from a research perspective) can be used as Personal Informatics tools has been argued for before in this workshop [1].

Insights derived from research oriented applications can potentially inform the design of personal applications, and vice versa. It is our position that in functional terms, just as the distinction between different protocols (e.g. ESM vs. diary) in research tools can be reduced to programmed behaviour within the same system, also the distinction between personal and in situ research informatics can be attributed to agency and role of the same system's user (e.g. researcher vs. private individual). Moreover, the same platform can be put to use for the purpose of both developing and researching/evaluating (in place and longitudinal) personal informatics applications. To this effect, we offer Tempest, our in situ data collection software stack, for consideration as a tool for prototyping personal informatics systems.

Tempest

Tempest is aimed at addressing the lack of an explicit paradigm for developing software for in situ data collection. It offers an approach to implementing such tools in incremental ways, by dispersing separated

concerns across the full spectrum of developer roles, from the software engineer to the end-user programmer, to the behavioural scientist issuing content. For further information, please refer to [2].

In terms of functionality, it is a modular software stack that enables researchers to serve users with interfaces they can interact with for the provision of data, or capture various kinds of implicit data, as can be made available by the platform at hand, e.g. sensor data, logs, etc. Finally, the collection of this data can be automatically triggered by events such as time, or entering a location. Through its design, Tempest achieves modularity, extensibility, and platform independence, and provides an editor for data collection protocols, which in diverse cases helps avoid programming the tool's behaviour altogether.

At the heart of the platform are the user-facing components and the basic client execution layer (Figure 1), which has the purpose of contacting the server, instantiating user-facing components, gathering explicit and implicit data, and submitting them back. It is a cross-platform HTML5 web application that can run stand-alone, or also be called and manipulated through what is commonly called a javascript bridge, from within a native application, in its SDK's web browser component. Moreover, it can run offline and encapsulates its own data handling without making this a concern of the developer. The actual interfaces that are served to the user can be any kind of javascript application. There's also the option of making these applications configurable through an authoring and monitoring GUI, wherein submitted data can also be monitored [3].



Figure 1: The system is layered as a stack of interchangeable components

Uses for Personal Informatics

Tempest can be readily adopted as a personal informatics application, featuring the ability to serve questionnaires to users, as well as non-standard widgets like a drawing canvas or a stopwatch. Further custom widgets are easy to add. However, we wish to indicate that it can also be used to iteratively develop better targeted personal informatics software, acting initially as a boilerplate prototype, and giving the ability to researchers to refine their designs based on data collected in place, from actual use.

While we are not in a position to report yet on a concrete case where tempest has been used for the prototyping of a Personal Informatics application, the following points list how its features can match the requirements that may come into play. Two roles are assumed here, the researcher, who configures the system, and the participant, who is the target/object of research.

- Interfaces served to users can be any kind of

application, from regular questionnaires for feedback, to mockups of interfaces for evaluation, to fully functional software that achieves a purpose related to collection of information. Some such are readily available (e.g.) questionnaires, and researchers are free to make their own without affecting other system components.

- Data is also transitted and processed in real time. Moreover, RESTful server APIs can make this data available to applications that need to consume it. Thus, the same user-facing javascript application that collects personal data can also consume and present data to the participant, e.g for feedback, visualizations, etc.
- Thanks to the nature of interpreted code, changes to the system's configuration on the user-facing layer can be reflected to what the participant sees in real time, without need for additional action (e.g. system updates). Therefore, questionnaires, mockups and interfaces can be iterated on rapidly.
- Tempest is platform independent in that it can run on web-browsers or in any native application that features a webView component, be that mobile or desktop. Although applications can be fully functional through Tempest, it is not restricting prototyping and development to take place within its own environment. While iterating on an application, researchers can choose to gradually phase solidified designs and functionality out of Tempest's environment into native code, and in the end rid of it completely.

Discussion

So far, Tempest has been employed in a few scenarios. It has been deployed as a standalone application for a diary study that recruited 20 participants for 4 weeks, who would log parameters around their consumption of video content. Also, for the evaluation of technology in the home environment in a single case study, used by a mother to log the sleeping routine of her infant. It has also been embedded within a tablet application intended for large deployment in the wild, for over time evaluation by its users. We shall be able to report on more and diverse use cases in the future that will help further mature the platform as an end-user development environment.

Conclusion

With the example of our own platform, we have attempted to support the preexisting argument for a more unified view of experience sampling and personal informatics tools. Moreover, we have discussed a tool,

which, applying the same principles, and because of modularity, can aid to the iterative development of personal informatics applications.

References

- [1] Experience sampling as personal informatics tools for persuasion. <http://personalinformatics.org/docs/chi2011/meschtscherjakov.pdf>.
- [2] Batalas, N., and Markopoulos, P. Considerations for computerized in situ data collection platforms. In *Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems*, ACM (2012), 231–236.
- [3] Batalas, N., and Markopoulos, P. Introducing tempest, a modular platform for in situ data collection. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*, ACM (2012), 781–782.