The UBhave Framework: Developing Dynamic Mobile Applications for Digital Behavioural Interventions

Charlie Hargood, Danius Michaelides, and Mark Weal
University of Southampton
cah07r@ecs.soton.ac.uk, dtm@ecs.soton.ac.uk, mjw@ecs.soton.ac.uk

Veljko Pejovic and Mirco Musolesi
Computer Science
University of Birmingham
v.pejovic@cs.bham.ac.uk, m.musolesi@cs.bham.ac.uk

Leanne Morrison and Lucy Yardley
Department of Psychology
University of Southampton
l.morrison@soton.ac.uk, l.yardley@soton.ac.uk

1. INTRODUCTION

Behavioural change interventions (BCIs) are used in behavioural psychology to elicit a desired behavioural change in a participant. These behavioural changes might be aimed at addressing a pupils behaviour in a classroom setting, or perhaps helping someone cope with the loss of a loved on. Many are used in the area of health promotion, such as smoking cessation, diabetes management, or weight loss. The interventions themselves usually contain a recognisable common set of features. These include collecting relevant information from the participant, setting goals and activity plans for the user, providing feedback and self-reflection, and the structuring of regular sessions designed to affect the change. Online interventions, termed digital behaviour change interventions (dBCIs) exhibit clear benefits, which have been articulated in [1], and include the comparatively low amount of research time and resources required, and the fact that they can be tailored and adapted more easily to fit the participants context and responses [2].

Smartphones have increasingly become the means by which people access the web and interact with digital content, this presents new opportunities and challenges for dBCIs. For example a weight loss intervention that may demonstrate success in a face to face environment, or through online materials at a desktop computer, might be even more effective if it could be delivered to an individual at vulnerable moments such as when the user is at a restaurant or shopping for food. In this work we present the UBhave Framework, a generic platform for creating and presenting mobile dBCIs. Our framework allows intervention authors to create a broad range of mobile dBCIs with custom content, tailoring, and functionality. These dBCIs are hosted on a central server and delivered to participants using an Android client which interprets the authors designs into a fully functioning Android application that records relevant data that is stored centrally for later export and analysis.

2. UBHAVE FRAMEWORK ARCHITECTURE

A generic approach to mobile dBCIs is a key goal of the UBhave framework. UBhave needs to abstract the details of intervention implementation, including the intervention dissemination, means of obtaining and processing mobile sensing data, monitoring intervention usage, and the interaction with the user, from the intervention designer. The framework needs to be usable by behavioural psychologists and therapists without mobile computing background. At the same time, the framework needs to be both highly flexible and feature-rich, and allow crafting of interventions that are context-aware, adaptable to user's behaviour and content-dynamic, as such interventions promise to improve therapeutic outcomes. Finally, UBhave aims to be scalable both in terms of the number of end users, as well as in terms of different dBCIs (digital Behaviour Change Interventions) supported at any given moment.

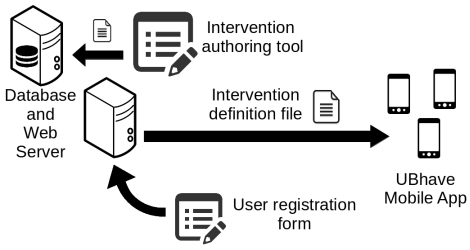

Figure 1: Overview of the UBhave framework.

2.1 Architecture Overview

We shift from the conventional means of architecting mobile dBCIs as static monolithic applications. Instead, we distribute our framework over the mobiles and one or more centralised servers, and decouple the intervention content, intervention's data collection directives and the logic behind the intervention interaction with the user, from the application that the users install on their phones. In such a system, the actual smartphone application serves as a vessel for a dBCIs designed by behavioural psychologists. The client becomes akin to a *browser for interventions*. Centralised servers provide intervention designers with easy access to gathered user data. The data can be collected on a large scale and dissected on various levels, and can include user-authored content, mobile context information as sensed by smartphone sensors, and application usage logs.

In Figure 1 we sketch the main components of the UBhave framework. The centralised server(s) hosts a secure database with user information and a web interface for user registration. An intervention authoring tool, which may reside on the same or a separate server, allows behavioural intervention designers to create a mobile intervention, spec-

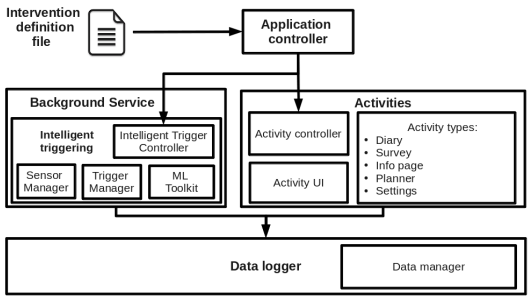

Figure 2: Android implementation of the UBhave mobile application.

ify everything from the text, picture and sound content of the intervention, to the rules of interacting with the users, and manage the groups of users that will have access to the designed intervention. The intervention specification is preserved in a database. The UBhave Android client can be freely downloaded by the potential users, who are upon starting it presented with a list of dBCIs that they are authorised to access. Selecting an intervention downloads an *intervention definition file* from the server, which is then interpreted by the mobile application. The definition file determines how will the application display the intervention content to the user, and which user-generated data will be collected and sent to the server. Finally, to make the intervention behaviour dynamic and adaptable to users' behavioural patterns, we implement a set of machine learning tools within the mobile application. The application builds an evolving model of user behaviour, and queries the models as instructed by the intervention definition.

2.2 Implementation

The intervention definition file is a JSON formatted output of the intervention authoring tool. The file contains intervention content, instructions on how the content should be displayed and optionally how the sensor data should be gathered. Through nested key-value definitions, the JSON file enables intervention content presentation through a layered hierarchy of menus, which is one of the ways of presenting dBCI content we identified during the initial codesing process.

Figure 2 presents the implementation of the UBhave framework Android application. The application hosts a persistent service and content display pages implemented as Android activities. The service maintains the logic for context sensing, user behaviour modelling, and notification management. Persistent operation is the key requirement of our framework, since we want dBCIs to interact with the user when needed, even if the user is not explicitly navigating to the application, and even if the interaction depends on the context. In our framework, the persistence is secured through the service that periodically saves the state of the application and the user behaviour models.

The mobile application is designed with a Model-Viewer-Controller (MVC) software architecture in mind. Figure 2 shows the hierarchy of controllers with the application controller on top. The *Application Controller* interprets a given JSON intervention definition file and delegates execution to other controllers as needed. Thus, the *Activity Controller* spawns UI components, as Android activities, while the *Intelligent Trigger Controller* controls notifications.

UBhave strives to balance the freedom that intervention designers have with the generalisability of the available components when it comes to UI design. We restrict the components to Info Page, Diary, Survey, Planner and Settings, all of which are implemented as Android activities. We support custom tailoring through predefined themes that an intervention designer can select, and HTML formatting that can be specified together with the content and interpreted by the mobile application. In future we plan to enable custom themes for further specification of the UI.

3. CONCLUSIONS

The UBhave framework provides a general model for representing mobile behavioural interventions that captures the key inherited requirements of legacy digital interventions, as well as the key affordances of the mobile platform. The framework architecture is a product of a co-design process between computer scientists and psychologists, through which we have been able to identify a number of important ways the support of mobile interventions differs from supporting PC-based interventions. The more reactive nature of pervasive technology requires in turn a more reactive model of intervention, and mobile intervention design needs to reflect this with less of a protracted planned session based approach and more of a context sensitive approach. In our framework we harness mobile sensing to infer the context, including location, physical activity, even users' internal states, e.g. interruptibility, and adjust the intervention accordingly. The framework's information display mechanism reflect this with less of a content page model and instead a model centred around context-sensitive notifications delivered on the mobile. Finally, UBhave differs from the existing experience sampling and mobile sensing frameworks as it taps into user's sensor data and intervention interaction data in order to tailor personalised interventions. In UBhave this tailoring is supported both through a structural system of functions and conditionals that depend on user-generated data, as well as through machine learning techniques employed to learn about user behaviour.

4. ACKNOWLEDGMENTS

This work was funded by the UK Engineering and Physical Sciences Research Council (EP/I032673/1: UBhave: ubiquitous and social computing for positive behaviour change).

5. REFERENCES

[1] F. Griffiths, A. Lindenmeyer, J. Powell, P. Lowe, and M. Thorogood. Why are health care interventions delivered over the internet? a systematic review of the published literature. *Journal of Medical Internet Research*, 8(2), 2006.

[2] M. Kreuter. *Tailoring health messages: Customizing communication with computer technology*. Lawrence Erlbaum, 2000.