Dimitra Gkatzia (Supervisor Helen Hastie)
Personal Health Monitoring

In this project, Stress Monitor (SM) is presented which is actually a mobile-based stress diary. The topic of Personal Health Monitoring is introduced and an overview of the related literature is given. Traditional stress diaries require much time from the user, they can easily be lost and they are not engaging. Stress Monitor aims to provide users assistance in filling the diary easier by using automated procedures where applicable and in reminding the users to fill the diary. It also aims to make them more engaged with the application. The methodology used in Stress Monitor development is the Waterfall model. In this document the systems requirements, design, implementation and evaluation are illustrated in detail. The comparison of SM with a web-based application indicate a remarkable inclination towards SM on account of the interface simplicity and friendliness, the ease of use, the small amount of information required, the portability that a mobile phone provides and the reminders that the user may want to set. Finally, the related ethical issues are discussed as well as directions for future research.

Keywords: Personal health monitoring, stress management, mobile application, eHealth, self management, Stress Monitor

Murali, Rajiv (Supervisor Andrew Ireland)

An Investigation into the Automated Generation of provably Correct Code from Formally Verified Designs

Industries involved in the development of safety critical systems are faced with conflicting demands of meeting both high assurance and productivity within the development of software systems. Auto-code generation has been an effective tool in bridging the stages of specification and implementation, thus improving productivity within software development. However, standards for safety-critical systems dictate a constant degree of verification through both these stages, and most commercially available auto-code generation tools have only targeted code generation and not the verification of the generated code. This project aims to investigate a novel approach to how properties defined in the specification of a system could be automatically translated and effectively utilized to provably correct code. To this end, we have focused upon the Event-B formal modelling notation, and the SPARK programming language. With this, the SPARK Examiner provides formal program analysis that helps eliminate security vulnerabilities and promote logical soundness within functionality of the system. Hence, by investigating this automatic generation of verifiable program code, an approach is taken towards meeting the standards of high assurance required in the development of safety-critical systems. The outcome of this project was the development of ESpark, an Event-B plug-in for automatic code generation to SPARK Ada. A series of successful experiments of the automatic code generation activity is detailed in the results of this paper. At this stage ESpark covers the sequential subset of Event-B system. Future development of the
auto coder could delve into new domains such as concurrent systems and could also look into targeting other program languages such as Java and C.

Murugaian, Deepak Raj (Supervisor Oliver Lemon)

Entering the Spoken Dialogue Challenge 2011 Using a Statistical Dialogue Manager

Recent advances in dialogue modelling, speech recognition and natural language processing have made it possible to build spoken dialogue agents for a wide variety of applications, (Walker et al. [1997]). They are used from simple Information access system such as Weather forecast information system to complicated agent interaction systems such as robot conversation systems. Even though the spoken dialogue systems are incorporated with a lot of advanced components, they are not consistent in noisy environments. It is very difficult to represent natural language as computer language so making them to interact in spoken language is still harder. This restricts in developing effective spoken dialogue systems by limiting their knowledge to a particular domain. For spoken dialogue systems, tracking a distribution over multiple dialogue states has been shown to add to the system’s robustness to speech recognition errors, (Williams [2010]). This document explains about participating in the Spoken Dialogue Challenge 2011 in which the task is to design a spoken dialogue system to provide bus schedule information for the general public of the Pittsburgh city, USA. We are using the AT&T Statistical Dialogue Toolkit (Williams ASDT, [2010]) to develop a mixed initiative spoken dialogue system which is based on Partially Observable Markov Decision Process. This thesis also describes the development of the system and its evaluation with similar systems.