5. Personalised Ambient Monitoring: Supporting Mental Health at Home

Evan MAGILL and Jesse BLUM
Computing Science and Mathematics, University of Stirling, Stirling, FK9 4LA, UK

Abstract. Many who suffer from Bipolar Disorder are keen to control their disease with as little external medical intervention as possible. Self help through websites, meetings, and questionnaires are commonly employed approaches. The PAM project has worked to help this process. It has endeavoured to form an ambient system of monitoring to provide objective feedback to bipolar sufferers. Particular effort has been made to allow the system network of sensors to be personalised and ambient, and operate without the need for a centralised resource. So a sensor system that embeds the processing of the sensor data has been developed. It allows the processing to be changed at run-time to allow personalisation and for changes in behaviour over time. This chapter describes the current status of the project; in particular it describes the rule-based system that the project developed, and an initial technical trial and its outcomes. The rule-based approach and the trial description should be of general interest to both technical developers and practitioners. The latter part of the chapter however is aimed more at the technical developer and focuses on the technical outcomes from the trial with a focus on the programmability aspects and addresses consistency issues that arise with such a flexible programming environment.

Keywords. Rule-based systems, run-time programming, personalisation.

Introduction

This chapter describes Personalised Ambient Monitoring. In particular it describes Personalised Ambient Monitoring (PAM) within the recent PAM research project. From the outset the PAM project has focussed on ambient monitoring for the mentally ill and in particular those with Bipolar Disorder (BD). The project has developed an infrastructure to allow a sensor network to be programmed at runtime to allow it to change its behaviour and response to sensor data. This allows the system to be tailored for individuals and to be altered over time as an individual’s mental state changes.

The chapter has two aspects. The first describes general experiences from the PAM project that should interest both practitioner and technology developer, while the second aspect continues with a more technical flavour. Readers should be clear that this chapter really describes a work in progress. Here it describes the first tranche of work, but the PAM consortium (of currently four universities) is actively planning further efforts. This chapter describes the project from the perspective of the University of Stirling.

Currently the project has developed an infrastructure for monitoring mentally ill patients in a home setting, and has tried this technology within a technical trial. It also has some very limited trial experience with patient volunteers but this is not explored within this book chapter. Clearly further work is planned in this direction. So this chapter is very much a “heads up” exploration of an exciting new application of (near) home monitoring of the mentally ill and the consequent essential technical developments. It is important for BD patients that the behaviour of the system can be adapted for individuals and easily altered to track their changes over time.

Although the term “monitoring” can sound imposed, the approach of the project was that of self-help. Indeed the system monitors with the explicit consent of the patient. Moreover it is expected to report its findings to the patient and so all the data and its implications remain with the patient. A patient may wish to share its findings with a clinician or informal helper, but that remains the prerogative of the patient.

The system described in this chapter aims to inform patients, particularly those using self-regulation, that their mental state is changing and may be moving towards a manic or depressed state. This allows a patient to be aware of these changes and take appropriate action.

The chapter begins by setting the PAM project in the context of current efforts in ambient monitoring. This is followed by a general description of the technical aspects of PAM that should appeal to a wide audience. A description of the technical trial and its ramifications follows. Finally the more technical aspects of PAM are addressed that should be of interest to technical developers. In particular it considers programming issues that arise from such a flexible programmable approach and highlights ways that the coherence of the system behaviour can be maintained.

1. Context and Setting

The last decade has seen the emergence of mobile and environmental sensor technologies being used to monitor health concerns at home and in real world (or ambulatory) settings. Many technologies have been applied to mental health care often under the auspices of ambulatory assessment [1]. Ambulatory assessment techniques record “ecologically relevant” data in real-time by monitoring subjects in-situ rather than in a clinical setting. These techniques have shown advantages over traditional retrospective recall-based methods. In particular ambulatory assessment avoids an inherent bias in the collected data when asking patients to remember and recollect experiences. In addition by collecting multi-modal data in-situ, the recorded data is context-rich and can be repeatedly assessed on-the-fly [2].

Many with BD endeavour to monitor their mental states for early warning signs of behavioural activity changes that alert them to impending manic or depressive episodes. Often such self-help includes the use of electronic or paper diaries to record mood [3]. These mood diaries allow BD sufferers to keep subjective records of their health.

The application of mobile and sensor network technologies to this type of monitoring brings with it novel development challenges related to system personalization. As there are differences in the way the disease manifests itself in different individuals, and the disease also changes over time for a particular individual, mental health care requires a higher level of system personalization and technological integration than is typical for other types of sensor networks. The types of sensors used, and their patterns of usage, must match transient subject states and be acceptable to the subjects. A detailed picture on the use of sensor networks for psychiatric health monitoring written by the authors can be found in [4]. In essence many projects had been proposed for general medical care or homecare but, to the best of the authors’ knowledge, the PAM project is the first to attempt to obtain activity signatures from the mentally ill using a network of environmental and worn sensors. In addition the existing projects had not shown how they respond to sparse datasets or integrate sensor data with self-monitoring reports.

The contribution of the PAM project is the introduction, within a mental health setting, of a rule–based approach embedded within the network. This permits dynamic and straightforward personalisation of network behaviour. It also supports additional equipment as and when it becomes available.

To provide a more flexible approach to ambulatory assessment in the PAM project, the activity signatures of individuals with BD are monitored through a set of sensors. The project constructed an infrastructure (PAM-I) composed of off-the-shelf and custom-built wireless sensors, and a rule-oriented programming architecture (PAM-A) to collect, process, and store data that can be related to models of care. The chapter describes both PAM-I and PAM-A in more detail later. The aim of PAM is to augment existing self-help strategies with objective information from the sensors. The driving philosophy is to process the data within the network close to the sensors. In addition the network components detect patient activity signatures and combine this information with electronic mood diary data. The proposition is that by monitoring activity signatures in naturalistic environments, the system may detect behaviour changes in mental health early enough to let sufferers and their care providers intervene effectively to prevent full-blown manic or depressive episodes.

A technical trial has been carried out to assess PAM-I and PAM-A reliability and acceptability to users. The emphasis to date has been on technical users, although a very limited patient trial has been performed. A longer term goal is to perform a limited clinical trial and to follow this with a full clinical trial.

The technical trial took place at four sites in the homes of project team members. This stage did not involve BD patients, but was performed by participants selected from the PAM project to exercise aspects of a subsequent patient trial. Ethical approval was obtained from the universities within the project. In addition to sensor data collection, interviews were conducted to gather the views of those being monitored before and after each trial.

This chapter describes the design and development of PAM-I and PAM-A as well as experiences and lessons from the technical trial. Later it discusses the programming issues that arose from the trial and describes candidate solutions. The earlier part of the chapter is expected to be of more general interest, while later on the chapter has a more technical feel.

2. Aspects of the underlying technology

The PAM system has been designed to allow the system to be easily programmed at run-time. The chosen approach of distributing rules is discussed in more detail later in the chapter for technical readers, but in essence it operates by distributing short sets of rules across the sensor network that control the overall behaviour of the system. As the rules can be distributed at any time the overall behaviour can be altered when required. By altering the behaviour in this manner, the way the system responds to the actions of a patient can easily be altered when necessary. This degree of flexibility is important in a mental health setting.

The rules are distributed throughout the sensor network onto a number of points such as mobile phones that can provide computing capabilities. It also has the potential to execute or obey these rules on smart sensors where a computer and a set of sensors are combined. The PAM system is designed for ambient mental health data collection and for reporting meaningful information to patients and care providers.

The system consists of two parts. The underlying *infrastructure* that provides the sensors, computer processing capabilities, and communications was named PAM-I to reflect its infrastructure role. The *architecture* that permits the distribution and execution of the rules was called PAM-A.

2.1. PAM-I

This underlying infrastructure provides a network linking wearable wireless sensor nodes, mobile phones, a personal computer (PC), and environmental sensor nodes installed in subjects' homes. PAM-I uses the Bluetooth protocol to provide wireless communication between a patient’s wearable sensor nodes and their mobile phone. Bluetooth is a short range wireless protocol often used to link PCs and keyboards. It is also commonly used to link mobile phones with wireless headsets. In PAM, Bluetooth is also used to communicate between a patient’s mobile phone and their in-home PC. A number of communications standards are used to link the environmental sensors and the PC; these include WiFi (IEEE 802.11b/g which is the common home wireless protocol used for home PCs, laptops, and game consoles), mains wiring (the power-line X10 protocol allows communications to be piggybacked on the domestic power supply), and Bluetooth. The PC also requires an Internet connection for secure off-site data storage.

Sensor nodes are composed of one or more sensors which are connected to a small embedded computer with communication facilities. The nodes may be mobile and require battery power, or they may be placed in a single location and be plugged into a wall power socket. In many cases the computing capabilities are restricted to keep a compact size and prolong the battery life. The technical trial involved participants carrying a mobile phone and two wearable devices. The first was an off-the-shelf GPS receiver and the other was a custom-made device with an accelerometer, light sensor and sound level sensor.

Each worn device is the size of a matchbox. The wearable nodes can be worn on belts or strapped to arms. The GPS units can simply be carried in pockets. Data is streamed from the sensors to the subjects’ mobile phones using Bluetooth. This allows the phones to collect data from the sensors and transmit the data to PCs for storage and analysis. This onward transmission to PCs may take place some time later as the phone holds the data until it is in range of the home PC. This allows a high degree of mobility for the user as they may not be at home or in wireless range. It also frees them from needing to use expensive mobile phones and any associated SIM contracts. Mobile phones control the sensor collection using rule-orientated applications. Additionally, momentary assessment of subject activities and moods can be collected through personalised on-phone questionnaires.

Various environmental sensors are integrated into PAM-I. These sensors are used to collect data about in-home subject activity. General home activity is monitored using light and sound level detectors, along with passive infrared sensors for monitoring how subjects are moving between rooms. Kitchen activity is monitored using a combination of micro-switches placed on kitchen cupboard doors and a wide-angle camera for monitoring many areas of interest (such as refrigerators and ovens). Images are processed to extract activity data, and only the activity data is stored to preserve subject privacy. In addition, there are sensors for monitoring television usage and sleep patterns. The television usage focuses on the rate a subject changes television channel.

2.2. PAM-A

A number of built-in applications were written to control the transmission, storage, and processing of the data. More technical detail is provided later in the chapter. Here the chapter simply highlights two capabilities that a patient can see: *PAM-Q* and *PAM-Pause*. Both take the form of mobile phone based questionnaires. Crucially these phone applications exploit the use of rules to allow each questionnaire to be personalised for an individual.

PAM-Q is a momentary assessment questionnaire application, and *PAM-Pause* is a second questionnaire to record the reasons why a subject is 'pausing' the monitoring system. Both the content of the questionnaires and the timings of the questionnaire presentation are configurable. The questions and answer options, along with settings that control the notification periods are stored in a knowledge base and are accessible for run-time modification without interrupting any other functionality.

3. Investigation

Members of the project team assessed the PAM system by installing the equipment and following the procedures that were developed in preparation for patient trials. This included two semi-structured interviews, and a raw data collection period. This form of investigation provided us with valuable insights regarding ambulatory assessment study design and technological development. The authors believe that tele-healthcare researchers and practitioners can gain considerable understanding by assessing their protocols first hand prior to studies involving patients which will resolve bias issues.

The study participants from the PAM project team did not have Bipolar Disorder but followed the trial procedures with the understanding that its use would be for Bipolar Disorder out-patients. To be included in the technical trial the participants provided ongoing informed consent and were free of serious health conditions that could have been aggravated by monitoring. Participants were allowed to withdraw from any part of the trial at any point in time. The participants were briefed about the data collection and processing prior to beginning the technical trial, and during a training session that followed device installation. They were shown how to record their answers to the questionnaires, and were familiarised with the importance of using *PAM-Pause* to distinguish deliberate pausing from missing data values arising from faults in the system.

3.1. Interview Procedures

Qualitative semi-structured interviews were designed to capture detailed points of view regarding the PAM project, its technology and monitoring in general. The entrance interview was focused on cultural factors as well as system acceptability and compliance issues. The exit interview was designed to review participation and to elicit thoughts and feelings about system acceptability, compliance and technological reliability.

3.2. Data Collection

Equipment was given to the four participants for on-body and in-home data collection, including wearable sensors, mobile phones and PCs. During the trials data was transferred from worn and environmental sensors and from mobile phones to the PC applications. The data was encrypted and transferred off-site for back-up and analysis. PAM technical trial data collection involved sensors being worn by the participants and placed in their homes. Customised mobile phones were provided to coordinate sensor communication and collect information about the participants' activities of daily life. The participants were asked to record their activities using a questionnaire application on the mobile phone. Continuous, discreet time-based sampling strategies were used to cover the variety of heterogeneous devices.

On-body device sampling rates were chosen to balance signal-processing concerns with the limitations of the wearable node processor and mobile phone memory constraints. The wearable nodes were not able to transmit accelerometer data and sound level data at the same time owing to hardware constraints. Instead, rules were customised to alternate their transmissions. One light meter reading was transmitted each second and transmissions alternated between five minutes of accelerometer data at 20 recordings per second and one minute of sound level recordings at 10 samples per second. The ordering and time values were stored in the rule-set at these rates for data analysis purposes. Raw data was streamed continuously from the wearable to the trial mobile phone where they were stored and forwarded to the subjects' trial PCs. The mobile phone also recorded data streamed from the external GPS node once every four seconds.

The data collection rates here may seem high. This relates to the dual nature of the data collection at this early stage of the work. On one hand the PAM architecture is designed to limit the data collection rates and process the data close to the sensors and so limit transmission and battery requirements, on the other hand as an experimental platform it was important to have the raw data for verification purposes. This dual aspect of the data collection was a particularly challenging aspect and did cause some compromises.

The mobile phone was set to display *PAM-Q* application notifications every hour. This sampling rate was determined in consideration of advice on time-based study design reported in [5]. The experiments showed that two questions in the questionnaire could be answered in less than 10 seconds if the application was already opened and in less than 20 seconds if the application needed to be launched. Notifications were personalised to fire every hour which minimised the self-report burden by fitting in to the participants' schedules. Notifications could be ignored by the participants, and they could enter questionnaire data at any time regardless of notification timing. Timestamps were recorded for the initial display time as well as for each response entry.

Two questions were available for answer on the mobile phone. Configurable response options were stored in the knowledge base. The response-set (happy, interested, anxious, angry, sad, ashamed, disgusted, other emotion, no emotion) for the question "How Are You Feeling?" was added to the rule-set and is derived from the basic emotions reported in [5]. The response options for the second question "What Are You Doing?" were personalised for each participant to correspond to activities that they were likely to engage in. These activities were discussed during the entrance interview. Activity options included a mixture of simpler activities (such as walking and sitting), and more complex activities (such as commuting or working). Multiple response options were selectable for each question.

Most of the environmental sensors performed continuous monitoring. However the passive infrared sensors (PIRs), micro-switches, bed sensor, and remote control usage sensor transmitted data on event-triggers. The minimum time between PIR sample transmissions was between 6 and 10 seconds. The camera transmitted one picture of the kitchen every 10 seconds to the in-home PC. The images were immediately processed to look for activities in areas of interest and only the activity data was stored. The activity data is simply noting activities within view of the camera. So in a kitchen setting the subject may have been in the fridge or has perhaps has used the sink.

In addition, ambient light and sound level samples were continuously streamed to the in-home PC over Bluetooth.

4. Lessons learned

The entrance interviews revealed some similarities in the participants' thoughts and feelings about project characteristics. All four participants were male and had a high degree of exposure to the trial equipment and technical trial protocols prior to beginning the technical trial. They all lived with at least one other housemate and sought permission of their housemates prior to installing the equipment. The participants had no or little prior experience of being monitored outside of the PAM project. They all expected to have an initial period of self-consciousness at the beginning of the trial that they presumed would lessen, but they didn't expect the monitoring to affect their general daily patterns of activity.

The entrance interviews also revealed distinct differences between the participants. For three of the participants, their feelings about the degree of monitoring in society ranged from generally negative, but acknowledging value for medical purposes, to wholly positive (no response was recorded for the fourth participant). Their views about how much time per day they thought was appropriate to spend interacting with the technical trial devices ranged from one minute per day or less, up to 20 minutes.

Participants highlighted reactivity, compliance and reliability issues to do with the technology and study design. Reactivity is defined in [1] to mean that the method of observation causes behavioural variance such as awareness, sensitisation, adaptation or coping strategies. For instance, three of the participants had some form factor issues with either the wearable unit or some of the environmental sensors.

Interesting lessons about monitoring were revealed during the interviews of participant 1. He had not been monitored before and was mildly concerned about exposing his life patterns to the other researchers. He required assurance that the data was anonymous. Other members of his household were concerned with camera data at first, but their concerns were alleviated when shown processed camera images, as these contained no identifiable information. Chapter 5 of [6] provides a detailed description of the image processing employed in PAM. This participant consented to the installation of all of the equipment described in section 3.2 except for the bed sensor. He allowed all of the chosen devices to record data through the full duration of the technical trial and, according to his exit interview, did not consider ending his participation early. His views surrounding the installation procedures changed during the study participation. During his entrance interview his only installation concern was about making sure that the sensors would not mark the paint on the walls. However, during his exit interview, the participant expressed additional concerns. He described a sense of loss of control owing to having three people taking part in the installation and a feeling that the installation took too long. As a result, in a later very small scale patient trial considerable effort was made to offer a professional manner and appearance, and there were always two installers present. The project's ethics-approval required that there was always more than one installer. In addition, explicit permission was sought to enter any space within the home. The equipment was now always fully configured before any installers entered a home.

Participant 1 and his housemates adapted quickly to most of the environmental sensors, but reacted in an unexpected way to the camera. He expected that the household would be self-conscious for the first couple of days but would not react further to being monitored. However, he reported in the exit interview that his household forgot about most of the monitoring equipment quite quickly except for the camera (because of its distracting bright blue LED).

Participant 1 found the system quite obtrusive and his compliance was affected by reliability issues. He reported poor mobile phone battery life, intermittent device disconnection problems, and form factor issues related to carrying three devices. He found the devices burdensome and worried about dropping them or leaving them behind. The wearable node irritated the participant because of its positioning on his belt as well as its intermittent communication connections. In response subsequent trials took steps to extend the battery life and reduce the impact of the camera. Connectivity was also improved.

Participant 1 reported positive feelings towards the mobile phone's form factor and screen quality, but he found the mobile phone questionnaire and transfer applications burdensome, and reported a fairly low compliance rate. He reported in the entrance interview that mobile application notifications every hour would be useful and an appropriate time setting, however he would have preferred the notifications to be triggered by changes of activity rather than timeout-based. In the entrance interview he believed that mobile questionnaire application interaction would be fine if it took less than one minute to complete per day. However, he reported that even though answering questionnaires was a quick process it was still overly intrusive and his compliance was low. Participant 1 also reported that he missed many on-screen notifications and preferred audio-based notifications. These findings are are in line with other research into interruptions such as [7] and [8].

The PAM team investigated mobile battery life performance. This showed that continuous data streaming over Bluetooth reduced the battery life by about 15% and file writing reduced it a further 25%. To examine the mobile phone's battery life the Nokia Energy Profiler version 1.2 was employed with different sensor configurations. It was expected that Bluetooth data streaming would be the largest power drain. Instead it was found that the internal GPS unit (which was subsequently disabled) and file writing to flash memory proved a much bigger power draw. Figure 1 shows consumption profiles. It compares tests starting 30 seconds into the recordings and running for three minutes. Profile 1 shows the mobile phone disconnected from the wearable node and not running any user applications. Its expected battery life was 9 hours at 0.41 Watts on average. Profile 2 shows data streaming from the wearable node to the mobile phone over Bluetooth, but without any on-phone data storage. Its expected battery life was 7 and a half hours at an average power consumption of 0.48 Watts. Profile 3 is the same configuration as profile 2 except that the data was being written to the phone's flash memory card. There was a lot more activity for profile 3 than there was for profile 2. Profile 3's expected battery life was 5 and a half hours with an average power consumption of 0.67 Watts. Profile 4 shows the power consumption for accessing the mobile phone's internal GPS. It had the highest impact on battery life with an average power consumption of 0.68 Watts and expected battery life of 5 hours.

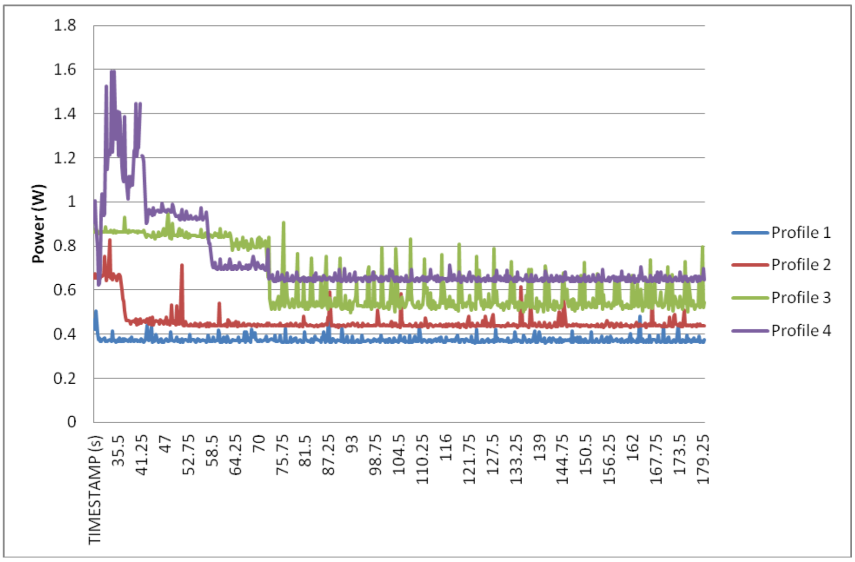

Figure 1. Power profiles for mobile phone – wearable node connection over Bluetooth.

The technical trial highlighted reliability and acceptability issues regarding on-body and in-home technology including mobile phone battery life, on-body gateway communications disconnection, on-body device form factor issues, and environmental sensor reliability issues.

More general issues also arise which are useful lessons for the next stage of PAM. The dual nature of the data collection where raw data and processed data is streamed together in places may affect the nature of the results. In retrospect this dual nature was not an ideal approach and further work may separate these streams. In addition much of the focus in the trial was technical and more emphasis needs to be given to the efficacy of the sensors used. At this stage the approach has been to use a broad range of sensors that echoed the direction given by the steering committee. The committee included members who were knowledgeable of Bipolar Disorder. The focus was on data likely to vary as a result of a subject's mood swings. However more work is required in the future to test the efficacy of particular sources of data.

A particular question raised by the trial related to the run-time programming ability of PAM; can it be ensured that the behaviour of the system remains coherent? The behaviour is frequently required to differ for individual users, and change over time for each individual. It is this particular thread that the remainder of the chapter considers. It is crucial component needed to allow personalized ambient monitoring to operate. Following a look at the more detailed aspect of the design, the chapter addresses this particular concern in rather more detail.

5. System Design

The remainder of the chapter (with the possible exception of the final conclusions) is by nature more technical and is less likely to be of general interest. Section 2 describes the more visible aspects of underlying PAM technology. Here, with the risk of some repetition, the text addresses the PAM technology exposing more of its hidden aspects.

As described earlier, the system consists of two parts: PAM-I and PAM-A. PAM-I is composed of off-the-shelf and custom built wireless sensors, and PAM-A that uses rules to control monitoring data streams and processing settings. Figure 2 shows the relationship between PAM-I and PAM-A. This shows sensors connected to nodes with limited processing capabilities communicating with devices such as smart phones and PCs. Rules residing on these devices control the monitoring system, dictating how and when to collect and store the data.

Section 2.1 describes PAM-I however Section **Error! Reference source not found.** below takes a more technical stance on PAM-A.

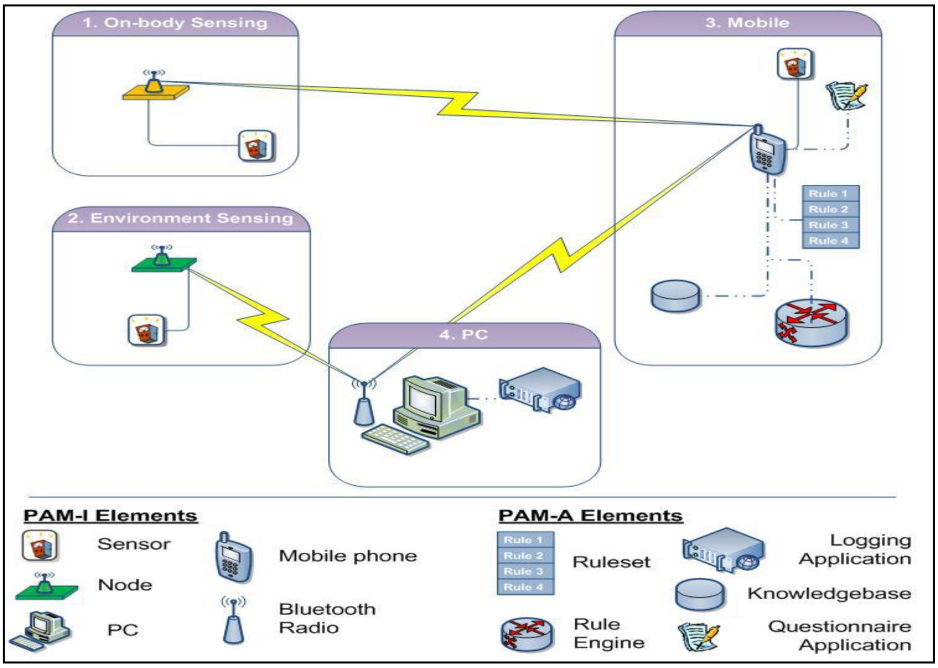

Figure 2. Graphical view combining PAM-I and PAM-A.

5.1. PAM-A

The Personalised Ambient Monitoring Architecture (PAM-A) is composed of custom applications to: handle inter-device network connections, control data streaming frequencies, record streamed data to persistent storage, and transfer data offsite for long term storage and analysis.

PAM-A is supported on the mobile phone and the home PC. Applications are programmed using a mixture of *Java* applications for device control and *Prolog knowledge bases* for rule processing to personalise the system. Data streamed from the devices are stored in XML documents conforming to the PAM sensor reading schema (PSR[1]). Sensor readings from heterogeneous devices can be stored in a single PSR file. The format is intended to:

- Allow readings from different devices to be interlaced throughout the file.
- Group readings into sets.
- Keep verbosity to a minimum in order to maximise battery-powered device lifetimes.

Four PAM-A applications were custom written for the mobile phones: *PAM-Gateway* can control data capture from the wearable units, *PAM-Transfer* performs mobile-to-PC data transmission, *PAM-Q* is a momentary assessment questionnaire application, and *PAM-Pause* is a second questionnaire to record the reasons why the subject is pausing the monitoring system. The applications were programmed using Java ME for Symbian S60 3rd Ed. [9] phones and M-Prolog [10].

The *PAM-Gateway* application provides software services to support connections to wearable sensor devices and to access internal phone sensors such as Bluetooth encounter monitoring. The rule-set is updated at run-time to provide flexible control over both the set of sensors that the gateway may communicate with, and the data streaming rates from the sensors. The application interfaces with a rule engine that is used to select appropriate settings on application start-up, and upon device reconnection. *PAM-Gateway* provides a pause and resume monitoring service and handles device disconnection-reconnection gracefully. The application accepts information from the multiple data sources and stores the information in files ready for transfer to the home PC using *PAM-Transfer*.

The *PAM-Gateway* application connects to the sensor nodes, controls their sensing characteristics, and stores readings from them. Device handler objects are instantiated at application start-up depending on rules in the knowledge base used to personalise the each user's devices. *PAM-Gateway* can control on-phone sensors (such as an accelerometer) and external nodes through Bluetooth communication. Device control rules contain handler names and configuration options such as data collection frequencies. Once instantiated, device handlers attempt to connect to their devices and forward control settings to them. Handlers maintain connections and respond to unexpected disconnections by automatically attempting to reconnect. Each handler registers a corresponding listener that listens for, and responds to, new data. Responses include recording data or performing additional data processing actions.

Data is written for each sensor into a PSR file corresponding to the *PAM-Gateway* session. The PSR format is designed to store readings from different sensors interlaced throughout the file, group readings, and keep verbosity to a minimum. Each reading belongs to a reading set describing the type of reading and the frequency that the reading was taken at. PSR files can be processed to aid in understanding of activity across the sensors.

---

[1] Normative and non-normative descriptions of the language and examples can be found online at http://www.cs.stir.ac.uk/~jmb/pam/readingXml/.

PAM-Q and *PAM-Pause* are questionnaire applications that can be personalised. Both the content of the questionnaire and the timings of questionnaire presentation are configurable. Questions and answer options, along with settings controlling notification periods are stored in the knowledge base and are accessible to *PAM-Gateway* for run-time modification without interrupting *phone service*.

Applications also run on the home PC to interface with the mobile phone and the environmental sensors, and to backup the data for long-term storage. A Java J2SE application runs on the PC for interfacing with *PAM-Transfer* as well as securely storing and backing up mobile and environmental data.

6. Programming Sensor Networks

PAM employs a rule-based approach to programming sensor networks. But why employ a rule-based approach? What alternatives are there? Currently, no consensus has been reached in the research community as to the best approach to programming sensor networks to meet these various issues. However, a rule-based approach to program sensor networks was chosen because of the advantages that this approach has over alternatives, such as distributed database models, agent-based programming and distributed virtual machines.

Rule-based middleware for sensor networks has been used in a number of projects. Sen & Cardell-Oliver [11] point out some of the advantages to rules-based sensor network programming. The authors explain that the programming and concurrency models are simplified compared with other approaches. They believe that program correctness is easier to prove, and that rule-based systems remain sufficiently expressive at high conceptual levels and power efficient.

Another benefit, according to Terfloth et al. [12], is that thinking about the system from an event paradigm applies better to sensor networks than thinking about the system using an imperative paradigm. Rule orientation, they argued, is a more natural way to express programs for sensor networks. In addition, Fei & Magill [13] showed that application developers using rule-based middleware are protected from complexities arising from tight real-world integration, network dynamics, and resource limitations. An interesting remaining area of concern is to what degree rule-orientated sensor networks have feature interactions, which will be explored further below.

Rule-based middleware for sensor networks has been used in a number of projects such as [11] - [14]. These studies show that the programming and concurrency models are simplified compared with other approaches. Furthermore they indicate that program correctness is easier to prove, and that rule-based systems remain sufficiently expressive at high conceptual levels. Also rule notations that employ an event driven paradigm find favour in sensor networks; whereas an imperative paradigm does not.

More generally rule-orientation is seen as a more natural way to express programs for sensor networks. It was pointed out by [11] that application developers using rule-oriented middleware are protected from complexities arising from tight real-world integration, network dynamics, and resource limitations. Rule-based systems have been built that allow the rules to be changed at run time [13]. This is very attractive for personalised systems that must change over time. Maintaining a consistent set of rules across the system, however, is challenging.

7. Maintaining rule consistency

In rule-based systems where rules may originate from a number of sources and end up being executed across a number of destinations, there is a strong possibility of the rules being inconsistent and causing behavioural conflict. This has been noted in [13], where they discuss the importance of detecting and resolving such conflicts. However that work did not address a method to do it. Instead the authors limited communication between peer sensor nodes and only allowed communication between individual nodes and a single server. By only accepting rules from a single trusted server, they avoided this issue as the trusted server employed meta-rules [15] to ensure conflict was resolved within the server and so conflicting rules were simply never distributed.

It is possible however to draw on a wider literature of programming conflict frequently described as Feature Interaction [16]. This topic was initially addressed in telephony, but has expanded to a wide range of domains experiencing program or control conflict; such as cars, lifts, internet services, and building control. The approach is to adapt the approaches developed in feature interaction to address rule conflict in (telecare) sensor networks.

8. Conflict Detection

In an ambient monitoring networked environment containing a dynamic collection of sensing and processing nodes that are attempting to detect unusual subject behaviour, network device conflicts are a common occurrence. For instance, features operating within and across devices could rely on synchronisation and concurrency patterns that may not actually arise owing to interactions between the devices and the rest of the network. Device conflicts reduce the levels of certainty that can be held in the care assessment data and the resulting conclusions that inform patients.

The remainder of the chapter describes an approach developed by the authors that operates on ambient monitoring services and networks.

8.1. Feature Interaction Problem

This work is inspired by research on the feature interaction problem as there are many similarities between this problem and rule conflict. A classic telephony example from the feature interaction literature involves the user Alice who is subscribed to the feature *Originating Call Screening* (OCS), screening out calls to the user Charlie. The user Bob is subscribed to the feature *Call Forwarding when Busy* (CFB), forwarding calls to Charlie when busy. A conflict can occur if Alice calls Bob when he is busy, because either the call from Alice would be forwarded to Charlie, thereby invalidating OCS, or else the call would be blocked, thereby invalidating CFB. In either case, the operation of one of the two features would be invalidated by the presence of the other.

It is important to note that this is not a traditional software error; it is not something that can be “debugged” and removed. Rather it is a conflict between the goals of the requirements of the two features. The goals of one feature simply conflict with the goals of the other. This cannot be resolved without changing the goals of one or more of the features.

Such issues are of course easier to solve if both features are developed within a single organisation. But in an open market this often does not happen. Indeed even in one large organisation, which is often the case with large software companies, the team developing the features may be located in different countries and subject to distinct management control. This issue of independent feature development is a common underlying theme in incidents of feature interaction.

It is also of note that the feature interaction problem has no formal definition. While rigorous descriptions are given, it does not have an agreed formal problem definition. This is as a result of it being a real practical problem in large software systems (initially telephone systems) that demanded attention. Hence it is very common in the literature to explain the problem by way of examples.

8.2. Rule conflict example

Here an example to highlight the feature interaction problem within a rule-based sensor network is presented. Consider a situation where a rule-based sensor system is employed to monitor a BD patient. A number of stakeholders may be involved in forming the rules: patients, clinicians, and possibly even family members. In addition rules for separate devices may originate from separate technical developers or companies. So the rules may well in fact originate from a number of separate sources.

Independently programmed device rules may, however, interact in such a way as to confound the overall goals. Consider a simple case that involves three sensor nodes to monitor a subject with BD. This scenario provides an example of how conflict can emerge from a set of interacting devices. The three sensor nodes used are: a *LocationMonitor* node (such as a wearable Global Positioning System (GPS) unit) that monitors where the subject is in the world, an *ActivityMon* node that monitors activities in which the subject is engaged (perhaps using a custom application on a mobile phone), and finally a *HomeMonitor* node that collects information from a variety of sensors in the subject's home. The nodes rely on each other to improve their performance, but such reliance may cause undesirable interactions.

In this example named *Brief Home Visit* (BVH), a set of rules attempt to save battery power by limiting the volume of data monitored when the context of the user does not require it. The individual device behaviours are controlled through rules on each device. So rule 1 of BVH, which is programmed on the *HomeMonitor*, stipulates that the *HomeMonitor* should only be active when the subject is located at home. Rule 2 for the *ActivityMon* defines that when the subject is at home the *ActivityMon* should be limited to only selecting from relevant home activities. Rule 3 also located on the *ActivityMon*, dictates that if the subject is performing the *travelling* activity, then only the start and end locations of the journey should be recorded in order to conserve node resources. Rule 4 (on the activity node) governs the length of the *travelling* activity as distinct from other activities. This value could be learned from user behaviour and could change in time. Rule 4 is set such that if the user enters the car, drives, makes a *micro stop* (such as buying milk at the convenience store) and arrives at a *macro stop* (for instance the gym, or place of work), it is all considered as part of the same single travelling activity.

This type of example can expose rule conflicts that may lead to unreliable behaviour. In this case, a particular type of rule conflict called *Missed Trigger Interaction* (MTI) can emerge. This can happen when the subject travels from home, returns home briefly, and then sets off again. When the subject initially leaves home the *HomeMonitor* is deactivated and the other two nodes enter their *travelling* states. When the subject returns home briefly (which was behaviour not considered by the rule authors), the *HomeMonitor* remains off (a trigger to turn it on is not sent by the *ActivityMon*) and so it does not capture any further abnormal behaviour patterns, resulting in a loss of data. When the subject leaves again, normal system data capture ensues. Figure 3 highlights this sequence of messages.

Conflicts such as these are often subtle. They are not necessarily obvious to anyone programming an individual device, especially when the programmers are unaware of the behaviours of all of the devices in the network. The system, therefore, should handle these types of conflicts. In other words the system should be able to automatically detect such conflicts.

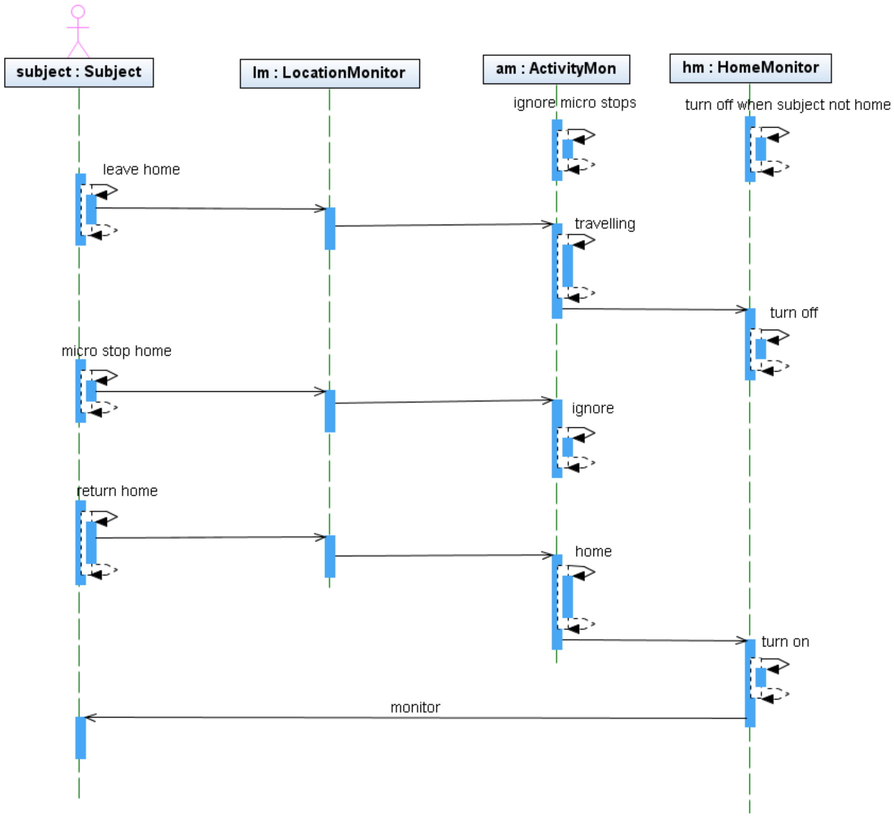

Figure 3. Brief Visit Home Sequence Diagram.

8.3. Types of Conflict

Various types of FI have been reported. An established form is Marples' FI taxonomy [17]. It consists of five types of interactions: *Shared Trigger Interaction* (STI), *Sequential Action Interaction* (SAI), *Looping Interaction* (LI), *Multiple Action Interaction* (MAI) and *Missed Trigger Interaction* (MTI). Here the chapter describes them briefly but more detail is available for the interested reader in a paper by Wilson et al. [18].

- *Shared Trigger Interaction* (STI) is where more than one feature responds to a single action or event and the resulting operation of at least one of the features is different from how it would have reacted had it been the sole responder.
- *Sequential Action Interaction* (SAI) is when a feature triggered in response to the actions of another. Such chaining of features may indeed be desirable, however it is still a form of conflict and may cause a feature to take action when it would otherwise have remained dormant.
- *Looping Interaction* (LI) can be considered to be a special case of SAI, whereby the operation of the chained features forms to a redundant cycle. While SAI can potentially be beneficial, LI can never be.
- When multiple features attempt to provide instructions or events for the same device, then the features have caused a *Multiple Action Interaction* (MAI). In some cases the interaction may be benign when both services send the same instruction to the device. However the interactions are often intolerable when the services send conflicting instructions.
- *Missed Trigger Interaction* (MTI) is when the operation of one feature prevents the triggering of another. In a sense the first feature is absorbing an action or event destined for the second feature. The second feature may be blocked waiting on a trigger, causing the feature to freeze or operate incorrectly. In practice MTI detection and resolution has proved very challenging to resolve.

8.4. Rule System

In order to detect conflicts between rules, an analytical rule system has been developed that can be used to understand what happens when multiple feature rules are triggered. The system analyses rule execution sequences to determine whether the rules lead to conflict. The framework ignores the contents of the triggering messages, the actions that arise from being triggered, and the semantic meanings of the features. Programs based upon the framework, resolve goals by loading the feature rules and then proceeding to check for interactions between every possible pair of features (including checking features against themselves).

Checking a pair of features involves two phases: initialisation and detection. The initialisation phase resets the environment by removing all values that have been set by the rules in earlier runs of the system. It then adds a number of time points (establishing a linear order amongst them) and initialises a message “fluent” that can be sent to the features. The fluent can be thought of as a value that is passed to the features, and as such could be considered as an event. The detection phase involves passing feature rules, time points and messages to a set of conflict detection rules. The conflict detection rules are then used to evaluate whether the feature rules are concordant or conflict, and to record evaluation results. The accuracy of the detection rules is crucial to the overall efficacy of this approach.

To ensure the detection rules are effective, a number of feature rules are analysed, looking for different types of conflict: such as *Shared Trigger Interaction* (STI), *Sequential Action Interaction* (SAI), *Looping Interaction* (LI), and *Missed Trigger*

Interaction (MTI). To do so, a set of scenarios based upon BD monitoring were developed for all five types of conflict.

Scenarios depict how features can conflict through a particular conflict type. For example, to test the MTI detection rules, a scenario describes a mobile phone containing a feature rule that delays the transmission of a message that would activate a home monitoring system. Such delays may be reasonable from a phone programmer's point of view to minimise bandwidth usage and maximise battery life. If the subject travels away from home, but returns home briefly, and then sets off again the home monitor would remains off because a trigger to turn it on would not be sent by the phone. This would lead to the system not capturing any abnormal behaviour during the brief return.

8.5. Conflict Detection Rules

Each of the conflict types are encoded as detection algorithms. These detection rules can be loaded into the analysis engine to check different device rules for conflict. Analysis involves checking whether messages hold at given points in time.

MTI arises when the operation of a feature prevents the triggering of the operation of another one. The second feature may get stuck awaiting its trigger which is delayed, thereby causing the feature to operate incorrectly or not at all. Detecting MTI can be accomplished by testing device rules sequentially to ensure that a common message holds before being passed to each of the tested rules. The message can be considered as a type of triggering mechanism that should remain in a consistent state during the whole scenario. Such an approach need not make any assumptions about the contents of the message, only that the content remains the same when passed to each of the device rules. It is possible for a rule to modify or destroy the message, if so then the message no longer holds, but is terminated at that time point.

The analytical framework evaluates a MTI concordance rule with arguments that consist of a pair of device rules, time points for the start times of each of the rules, and the message. The message initially holds prior to being passed to the first rule. Device rules conflict if the first rule message terminates the message prior to the execution of the second rule.

STIs occur when the antecedents of multiple rules are satisfied such that they each perform actions in response to the same triggering event, and the operation of one or more of the rules are different from how it would have reacted had it been the sole responder. STI detection begins by loading arguments that consist of a pair of device rules, but ignores the time points and the message arguments. Testing for STI can be accomplished by querying a rule representing a feature, then resetting the query environment, querying a second rule, then querying a second instance of the first rule. If a check of the initiated actions from the first instance of the first rule does not match the second instance's initiated actions then the first and second rules conflict by STI.

SAIs occur when the operation of a device rule is triggered in response to the actions of another device rule. SAI can be detected by testing to determine if a device rule performs an action that leads to actions being performed by a second rule. This can be accomplished by running rules sequentially within the framework and checking for sentences that describe actions that will be performed as a result of the firing of the two rules. The analytical framework detects SAI by firing rule 1, then resetting the environment and firing rule 2, followed by rule 1 again. The actions from each of the rule 1 firings are compared and SAI is detected if they are different.

LI occurs when one rule triggers another, which in turn causes the first one to be re-triggered. LI, therefore, is a special case of SAI that can be defined as SAI leading to the triggering of the first rule's actions. This can be detected by performing SAI checks on the rules and examining the output for cases where two rules have SAI regardless of whether they are the first or second rule. In such cases the device rules will each perform actions that lead to the other being performed. If rule A causes an action that fires rule B and rule B causes an action that fires rule A, than a loop has occurred.

9. In conclusion

This chapter considers the infrastructure necessary to support ambient monitoring of Bipolar Disorder patients. It presents an architecture that gives the degree of run time programmability that is required to provide a personalised solution that can be programmed to meet an individual's needs and can change over time to reflect how the condition changes over time for that individual.

The chapter started by highlighting how the PAM system has been trialed to provide ambient mental health data collection and so preparing the way to report meaningful information to patients and care providers. The system consists of two parts. Firstly an infrastructure (PAM-I) composed of off-the-shelf and custom built wireless sensors, and secondly a programming architecture (PAM-A) that uses rules to control monitoring settings and stream data for processing.

With a view to ease of programmability, the issue of rule conflict was investigated and the chapter concludes with a tried solution to ensure rule coherency.

Work is in progress to develop and extend the initial trials to a larger more comprehensive and extensive evaluation. However studies reported here have been essential in forming a useful and flexible home based personalised monitoring platform.

10. Acknowledgements

This work was carried with the support of the EPSRC funded project EP/F003684/1 (Enabling health, independence and wellbeing for psychiatric patients through Personalised Ambient Monitoring (PAM)). The PAM project is a collaborative project between the Universities of Stirling, Nottingham and Southampton.

The authors would like to thank all of the PAM research team for their help and support.

References

[1] J. Fahrenberg, Assessment in daily life. A Review of Computer-assisted Methodologies and Applications in Psychology and Psychophysiology (years 2000 – 2005) website, *http://www.ambulatory-assessment.org*, last accessed May, 2012.

[2] U.W. Ebner-Priemer, and T.J. Trull, Ambulatory Assessment - An Innovative and Promising Approach for Clinical Psychology, *European Psychologist*, **14** (2009), 109-119.

[3] S. Lane, N. Heddle, E. Arnold, and I. Walker, A review of randomized controlled trials comparing the effectiveness of hand held computers with paper methods for data collection, *BMC Medical Informatics and Decision Making*, **6** (2006), May 2006.

[4] J Blum and E.H. Magill, M-Psychiatry: Sensor Networks for Psychiatric Health Monitoring, *PGNET 2008: The 9th Annual Postgraduate Symposium on The Convergence of Telecommunications, Networking and Broadcasting.* Liverpool, June, 2008.
[5] U.W. Ebner-Priemer and G. Sawitzki, Ambulatory Assessment of Affective Instability in Borderline Personality Disorder - The Effect of the Sampling Frequency. European Journal of Psychological Assessment **23**(4) (2007), 238-247.
[6] J.D. Amor, Detecting and monitoring behavioural change through personalised ambient monitoring, Doctoral Thesis,(2011).
[7] B.P. Bailey, J.A. Konstan, and J.V. Carlis, The effects of interruptions on task performance, annoyance, and anxiety in the user interface. INTERACT Vol. 1, (2001), 593–601.
[8] P.D. Adamczyk, S.T. Iqbal and B.P. Bailey, A method, system, and tools for intelligent interruption management. Proceedings of the 4th international workshop on Task models and diagrams - TAMODIA '05, 123, (2005).
[9] "S60 platform overview", website, *http://www.developer.nokia.com/Devices/Symbian/*, last accessed May, 2012.
[10] "mProlog: Lightweight PROLOG Engine", website, *http://www.cs.uu.nl/3apl-m/mprolog.html*, last accessed May 2012.
[11] S. Sen, & R. Cardell-Oliver, R. A rule-based language for programming wireless sensor actuator networks using frequence and communication, *Proceedings of Third Workshop on Embedded Networked Sensors (EMNETS)*, **1** (2006).
[12] K. Terfloth, G. Wittenburg, & J. Schiller, FACTS - A Rule-Based Middleware Architecture for Wireless Sensor Networks. *Proceedings of the First International Conference on COMmunication System softWAre and MiddlewaRE (COMSWARE '06)*, (2006).
[13] X. Fei, & E. Magill, Rule Execution and Event Distribution Middleware for PROSEN WSN, *Second International Conference on Sensor Technologies & Applications, SENSORCOMM'08*, (2008) 529–551.
[14] M. Zoumboulakis, G. Roussos, & Poulovassilis, A., Active rules for sensor databases, *ACM International Conference Proceeding Series*, **72** (2004), 98-103.
[15] G.A. Campbell and K.J. Turner, Goals and Policies for Sensor Network Management, *Proc. 2nd Int. Conf. on Sensor Technologies and Applications*, (2008), 354-359.
[16] M. Calder, M. Kolberg, E.H. Magill, & S. Reiff-Marganiec, Feature interaction: a critical review and considered forecast, *Computer Networks*, **41**(1) (2003), 115–141.
[17] D. Marples, Detection and Resolution of Feature Interactions in Telecommunications Systems During Runtime. Ph.D. thesis, University of Strathclyde (2000).
[18] M. Wilson, M. Kolberg, & E.H. Magill, E.H, Considering side effects in service interactions in home automation-an online approach. Feature Interactions in Software and Communication Systems, **9** (2007), 172–187.