eWatch: Context Sensitive System Design Case Study

Asim Smailagic, Daniel P. Siewiorek, Uwe Maurer, Anthony Rowe, Karen P. Tang
Carnegie Mellon University
{asim, dps, umaurer, agr, kptang} @andrew.cmu.edu

Abstract

In this paper, we introduce a novel context sensitive system design paradigm. Multiple sensors/ computational architecture, in the form of our eWatch device, is used to infer the activities that the system is encountering, and can provide a platform for context-aware computing. We created an eWatch prototype that senses user activities and notifies them when important messages have arrived. An accelerometer and microphone provide inputs to a model of interruptibility. A vibration motor for tactile feedback and two ultra bright LEDs for visual feedback provide user notification through different vibration patterns and colors. eWatch is transparently integrated into the user's environment, and communicates via Bluetooth. This new class of integrated systems underscores the need for new forms of regularity, constraints, and design structure. Our results indicate the power of our method to accurately determine a meaningful context model while only requiring data from our eWatch device.

1. Introduction

In this paper, we introduce a novel context sensitive system design paradigm. Multiple sensors/ computational architecture, in the form of our eWatch device, is used to infer the activities that the system is encountering, and can provide a platform for context-aware computing. We present a dual purpose, sensing and notification platform, for context aware wearable systems, as defined in [1], [2], and [3]. The design was driven by a vision of future electronic design that includes application-domain-specific configurable logic and sensor components. As a testbed, we designed eWatch, an electronic watch that senses user activities and notifies the user when important messages, including emails, have arrived. eWatch employs an accelerometer, a light and temperature sensor, as well as a microphone to sense user activities. In addition, cybersensor data, such as user calendar data from a PDA are input to a model to estimate user interruptibility levels. eWatch communicates with a PDA using Bluetooth. A vibration motor for tactile feedback and two ultra bright LEDs for visual feedback provide notification to the user through different vibration patterns and colors. User studies helped to identify appropriate notification schemes for mobile and office settings. Based on user context, the system uses incoming email priority levels to determine whether and how to notify the user. eWatch is transparently integrated into typical human activity patterns.

Our approach to context sensing eliminates the calibration and hand-tuning problem. After a brief period of learning from data gathered by sensors and from databases, the system determines which patterns of data correspond with particular user behaviors. This is far more flexible and robust than predetermining, for example, that the system shall decide that a user is in a conversation by looking only at data from a microphone, as it allows the system to automatically take advantage of information provided by multiple sensors that a would not have intuitively seemed relevant. Furthermore, it simplifies the notion of context, requiring the system only to identify previously seen patterns, rather than having to label every user state. Our results indicate the power of our method to accurately determine a meaningful context model while only requiring data from our eWatch device.

eWatch was designed to provide a rich set of input and output modalities that could be easily explored in a pervasive computing environment without any on-board computational restrictions. This introduces new capabilities, as work in previous wrist watch computer systems, such as in [4], focused on the engineering challenges in building such a miniaturized device. These new capabilities and the context-specific approach itself represent an important advancement in the state-of-the art.

2. System architecture

eWatch is a thin client that collects sensor data and can send user notifications. eWatch was designed to be a wearable testbed that can also use off-board computational facilities. eWatch utilizes a wireless Bluetooth connection to a computer that can analyze the sensor data and appropriately activate the different eWatch actuators up to 15 meters away. As seen in [5], ease of development frequently hinders research beyond the immediate engineering challenges. Using a desktop computer or a portable PDA allows rapid

Proceedings of the IEEE Computer Society Annual Symposium on VLSI
New Frontiers in VLSI Design
0-7695-2365-X/05 $20.00 © 2005 **IEEE**

development and easy deployment of different test scenarios. The reduced computational requirement on the eWatch also allows it to fit into a small package, as shown in Figures 1 and 2. We used the EAGLE tool for electronic design, and Fused Deposition Modeling (FDM) and Stereolithography (SLA) for packaging.

The eWatch architecture consists of three major components: the main controller board, the Bluetooth module and the host computer. The main controller board communicates over RS232 to the Bluetooth module which in turn connects to the host computer. The eWatch architecture can be seen in Figure 3. Note that the Bluetooth module and the main board are physically separated into two printed circuit boards.

The main controller board shown in Figure 4 for the eWatch is responsible for all sensor integration as well as controlling the LCD display and actuators. The main processor on the eWatch is an 8Mhz Microchip PIC18F2320 with 512 bytes of RAM, 4096 bytes of program memory and 256 bytes of EEPROM. The microcontroller includes eight 10-bit ADCs, as well as a variety of communication interfaces.

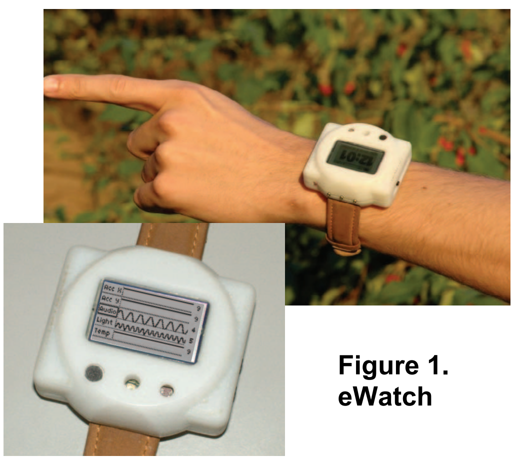

Figure 1. eWatch

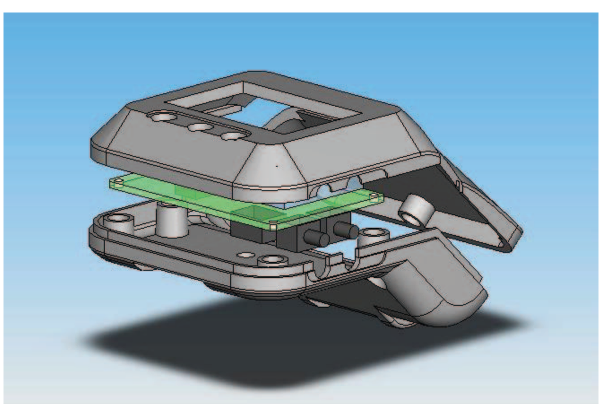

Figure 2. eWatch package

eWatch collects user state information from a light sensor, temperature sensor, microphone, and dual axis accelerometer. The light sensor is calibrated to return a 10 bit value that covers a range between total darkness and direct sunlight. The temperature sensor is attached to a thermally conductive metal plate that is pressed directly against the user's skin and is sensitive to .1ºC changes over a range of 0 and 100ºC. The microphone data is processed by the main PIC processor and is used to detect the loudness of ambient sound. In order to minimize size and power, the microphone uses a MAX4061 single chip amplifier with a variable gain set to 1000. Acceleration is sensed using an ADXL202 MEMS accelerometer that measure two axis with +/-2g of range. Three push buttons are distributed around the outside of the controller board in a fashion that emulates the standard configuration of a digital watch. All sensor data is packaged and transmitted over Bluetooth to the host computer 20 times per second.

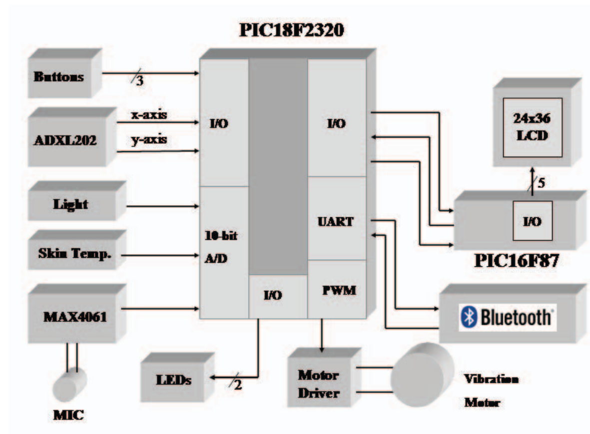

Figure 3. Hardware architecture

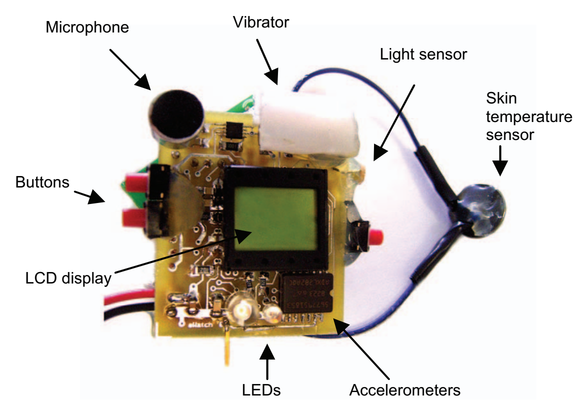

Figure 4. eWatch controller board

The eWatch supports both tactile and visual output. The tactile output is generated by a small vibration motor similar to that found in cellular

Proceedings of the IEEE Computer Society Annual Symposium on VLSI
New Frontiers in VLSI Design
0-7695-2365-X/05 $20.00 © 2005 **IEEE**

phones. Using the PWM generator on the main processor it is possible to generate many levels of vibration intensity. By mixing different durations, intensities and pauses, the eWatch can generate a vast variety of patterns, many of which are well beyond human distinction. Two ultra bright LEDs could also be alternated and individually modulated yielding a similar level of variety. The 24x36 pixel LCD is driven by its own PIC16F87 microcontroller. Images and time are sent from the main processor to the LCD controller using a custom bus. The LCD controller has an additional red and green LED that can control the backlight of the screen.

The main board, the Bluetooth module and a 7.2 volt lithium polymer battery are packaged together in a padded leather band that fastens around the user's wrist with Velcro. The device on average consumes 0.5 watts which given the lithium battery's capacity lasts approximately 12 hours. The eWatch controller board is 33mm by 45mm and the final system weights 85 grams without the battery.

3. System Architecture

The eWatch software architecture is a distributed client/server system, with the three main components: eWatch, PDA and server, as shown in Figure 5. All components communicate via wireless network protocols. There is a Bluetooth link between eWatch and the PDA, and 802.11b between the PDA and server.

The server provides the PDA with information about incoming emails, specifically the priority of the emails. The server has a database with information on how to connect to the users' email accounts. Periodically the server queries these email accounts for new emails. The prioritizer analyzes the emails and assigns them a priority. The processed emails are stored into the user's email queue for later retrieval by a client, in this case a PDA, which queries the server for the user's prioritized emails. To execute the query we defined an ASCII protocol on top of TCP/IP, which allows the PDA to authenticate, query, and receive new emails from the server.

The software on the PDA for the email notification consists of the: notification manager, interruptibility module and decision making module.

The client side of the **eWatch driver** interfaces with the eWatch using Bluetooth. It defines an API to control the LEDs, LCD screen, and the vibration motor. The current values of the sensors and state of the buttons can be queried by the driver.

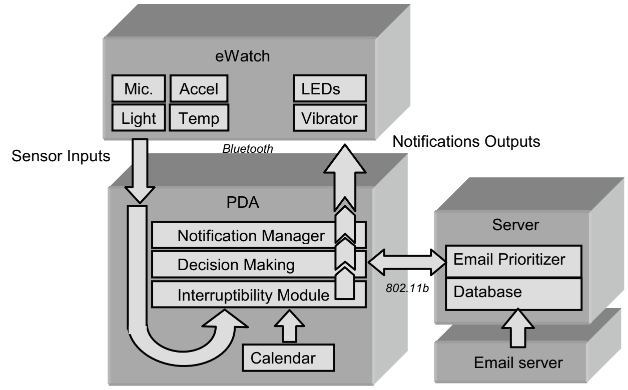

Figure 5. System Architecture

The **notification manager** sends different types of notification patterns to the eWatch. These patterns define how the LEDs and the vibrator motor are activated during a notification. The patterns range from simple to complex. For example one basic pattern might be a short activation of the vibration motor. A more complicated pattern would incrementally increase the vibration intensity while blinking the LEDs with varying frequencies.

The **interruptibility module** collects the sensor data and the information from the calendar to determine the user state. This state represents the level of user interruptibility.

The **decision making module** combines the information about interruptibility with the priority of the emails. Based on the notification modality matrix, this module decides which pattern is appropriate to notify the user about the incoming email.

An interface to connect the eWatch to our context aware machine learning system called ARIUS was developed. It enables input of the sensor values to the unsupervised learning module and clustering algorithms of the system [2].

4. Experiments

The interruptibility and decision making pair of modules, shown in Figure 5, creates a two-tiered matrix, which maps input variables of email priority and interruptibility to specific output modalities of the eWatch.

The eWatch architecture determines interruptibility based upon a sampling of sensors. In our current prototype they include an ambient microphone to detect volume levels of the ambient noise within the environment, light sensors to detect shades of daylight exposed to the eWatch, a two-dimensional accelerometer to detect motion characteristics, and a body temperature sensor.

Based on our analysis, we produced the interruptibility matrix shown in Table 1. The columns indicate the appropriate interruptibility levels which correspond to the input variables listed on the left side of the table. The first row depicts the event's priority level on the user's calendar that would be required to interrupt the user in the three user states. A priority level of one is the highest and most urgent, while four is the lowest. Some of the columns are further subdivided into different categories that describe different situations that fall under that category. For example, the user may be somewhat interruptible in lecture hall (H), discussion (D) or while at the library (L). The subsequent rows list the sensor values for each physical situation (lecture, discussion, library, test (T), presentation (P)). Foreground noise quantifies short bursts in loudness which correlates to conversation. Low and high refer to the loudness of the foreground noise. The background noise describes the persistent noise level that is averaged over a long period of time. In this case, more than just intensity should be considered. The first background noise sub-row is the intensity. The next row should be used if the number of noise producing sources can be identified. The final row should be considered the distance of the background noise.

The last sensor input row shows the amounts of accelerometer activity required for the user to enter each interruptibility state. The "O" stands for a low level activity, an "X" indicates a moderate level of activity, and a triangle indicates a high level of activity.

The primary purpose of the interruptibility matrix is to parse the input values of the incoming sensor

Table 1. Interruptibility Matrix based on experiments

	Context						
Sensors and data sources	Highly Interruptible		Somewhat Interruptible			Not Interruptible	
Cybersensor (Calendar) priority	Don't Care		1 – 4 H	1 – 4 D	2 – 4 L	1 T	1 P
Fore ground noise	Don't Care	Don't Care	low	high	low	low	High
Back ground noise	high	Don't Care	high	high	low	low	low
	multiple	Don't Care	single	single multiple	Don't Care	Don't Care	Don't Care
	Don't Care	Don't Care	far	near	Don't Care	Don't Care	Don't Care
Accel.	O	Don't Care	X	X	▲	X	O

data. eWatch effectively closes the loop between the input and output modalities.

4.1 Context Sensitivity Study

Our hypothesis is that it is possible to combine low cost sensors with machine learning algorithms to infer context with mobile electronic devices.

We consider the user's state to be a variety of different context qualities such as location and activity that can be estimated by using wearable sensors. We observe that context does not require a descriptive label to be used for adaptivity and contextually appropriate response. This makes an approach towards completely unsupervised learning feasible. By unsupervised learning we mean the identification of the user's context without requiring manually annotating current user states (i.e. without external supervision).

Unsupervised machine learning techniques are used to independently cluster sensor quantities and associate user interactions with these clusters. The use of this discretization enables learning from observations about the user. Each time an interaction is observed, it is interpreted as a labelled example which can be used to construct a statistical model for context-dependent preferences.

4.2 Study Results

To evaluate our unsupervised machine learning approach for context identification, we performed the following experiment. Two subjects wore the wearable sensor device over the duration of several days, personally annotating their state – i.e. working in the office, being in a meeting, commuting – by pushing a button and additional note-taking. The labeling and granularity of manual annotation corresponded to the clustering results and granularity of the automatic classification. The subjects were required to mark time-stamps when they changed contexts and verbally classify the corresponding state. In summary, over 240 hours of studies were done by recording one sample per minute.

Data was analyzed in blocks of varying lengths, including a variety of different contexts. An initial clustering with on average 15 clusters was done per block. The data was processed by our classifier and the resulting clustering was manually compared with the data annotation. We examined whether equally annotated states correspond to single or separate clusters. Figure 6 displays the performance of our method on a 20 hour data sample collected during that experiment. The following abbreviations are used: L –

Proceedings of the IEEE Computer Society Annual Symposium on VLSI
New Frontiers in VLSI Design
0-7695-2365-X/05 $20.00 © 2005 **IEEE**

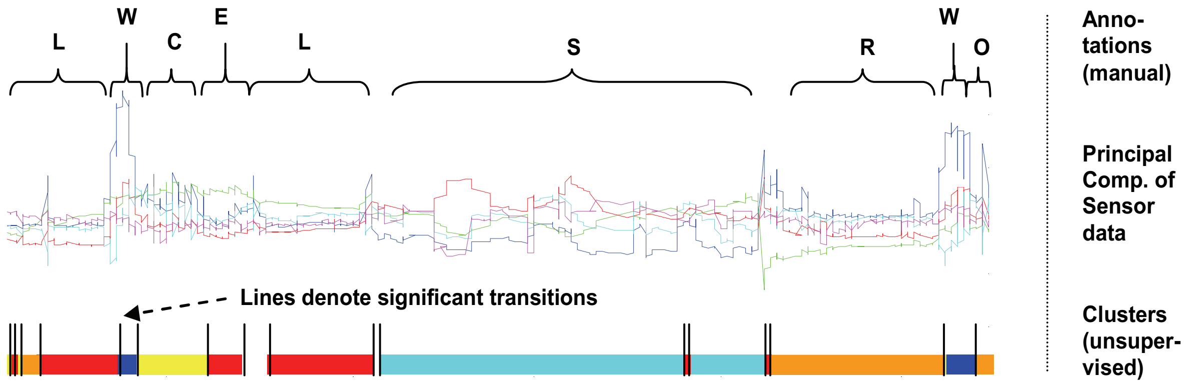

Figure 6. Analysis of the clustering – annotation correspondence

working on laptop, W – walking, C – cooking, E – eating, S – sleeping, R – reading, and O – office. Our method could compete with the subjective context classification. The data from the experiment was input to our unsupervised algorithm which showed comparable classification performance for the number of clusters and the transition locations. The state transitions determined by our classifier closely matches the number of timestamps recorded by the subjects.

4.3 Context Sensitive Output

The current eWatch prototype is equipped with a blue and red LED light (for visual feedback), as well as a vibrating motor (for tactile feedback). A user study was conducted on the various combinations of notification modalities. The LEDs and vibrating motor were tested on three parameters: intensity, duration, and pattern of activation. After each notification, the participant is requested to acknowledge the pattern by pressing a button on the watch. The time between the end of the notification pattern and the time to when the eWatch button is pressed is referred to as the "response time." Based on the subject's response time, we aimed to determine which notification pattern is most effective.

We performed a preliminary study, choosing a few patterns to send to the users while they conversed. The two relevant observations we derived from the study are that signals must be made distinguishable in at least two or more parameters, and that there should be a pause between changes in intensity.

Based on this new information, we ran a complete study. One group of participants had a primary task engaged in oral conversation. Another group was engaged in a computer-related task. In this iteration of the study, we chose 14 notification patterns to send to the participant's watch based on the observations we made in the prior study. The patterns are listed in Table 2. Each pattern had a total duration of 10 seconds. By measuring the response time (in seconds), we obtained the results in Figure 7.

Table 2. Notification patterns used in user study

#	Visual	Tactile
1	None	Gradually increasing intensity & frequency
2	None	Single medium intensity
3	None	Long medium intensity, pause Long light intensity, pause Long heavy intensity
4	None	Three long medium intensity, steady frequency
5	Red, steady frequency	None
6	Blue, steady frequency	None
7	alternating Red & Blue	None
8	Fading Red	None
9	Fading Blue	None
10	Red, pause Blue, pause Blue, pause Red	Long medium intensity, pause Short medium intensity, pause Short medium intensity, pause Long medium intensity
11	None	Gradually increase intensity, steady frequency
12	None	Single short low intensity
13	None	Single short high intensity
14	None	Three medium low intensity, steady frequency

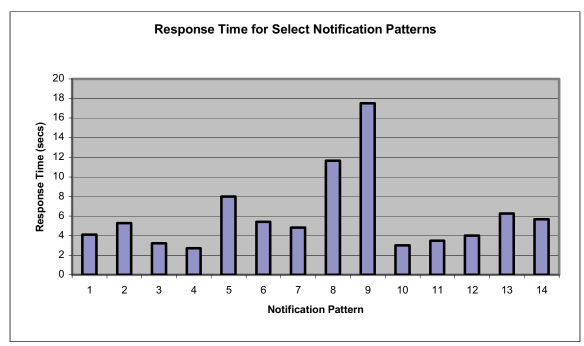

Figure 7. Response Time for User Study Notification Patterns

5. Conclusion

This paper presented a dual purpose sensing and notification platform for context aware wearable systems. In particular, we address activity sensing and notification mechanisms. eWatch senses user activities and notifies them when important messages, such as emails have arrived. An accelerometer and microphone provide inputs to a model of interruptibility. A vibration motor for tactile feedback and two ultra bright LEDs for visual feedback provide user notification through different vibration patterns and colors. Our experiments identified appropriate notification schemes for mobile and office settings. The system combines an interruptibility measure and email priority level, and cybersensor data to decide how and when to notify the user.

We created an interruptibility matrix, which is based on email priority and interruptibility as inputs, and tactile/visual modalities as outputs. Our experiments and their results are also presented in the paper.

Building novel context sensitive platforms, such as the eWatch which are truly integrated systems with full input and output interfaces to the physical environment is a challenging task. It is quite apparent that new forms of regularity, constraints and design structure must be employed to make such systems economically feasible. Even implementing large digital systems in deep submicron silicon has presented numerous challenges for ASIC feasibility. As we add sensors and other heterogeneous system level components, the cost and design time required for test, manufacturing, verification, redesign, etc., will only increase. For this reason it is apparent that existing application-specific IC design methodologies will not serve as a vehicle for the next generation of system-on-chip integration, and will soon be limited to a narrow class of design problems just as full-custom integration is today.

To support this class of systems, a new methodology for context-specific IC design must be created that includes application-domain-specific configurable logic and microelectromechanical (MEMS) sensor components. This will greatly benefit the available MEMS and sensor technology, presenting new opportunities for integrating circuits and sensors much more tightly, and thereby establishing new capabilities that would otherwise be infeasible.

6. Acknowledgements

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHC030029 and the National Science Foundation under Grant Nos. 0205266 and 0203448, and the Pennsylvania Infrastructure Technology Alliance.

7. References

[1] Siewiorek, D., Smailagic, A. et al. “SenSay: A Context-Aware Mobile Phone”, Proc. ISWC 2003, White Plains, New York.

[2] Krause, A., Smailagic, A., Siewiorek, D., “Unsupervised, Dynamic Identification of Physiological and Activity Context in Wearable Computing”, Proc. ISWC 2003, White Plains, New York, pp 88-97.

[3] Smailagic, A., Siewiorek, D. P., “Wearable and Context Aware Computers: Application Design,” IEEE Pervasive Computing, Vol. 1, No. 4, 2002, pp. 20-29.

[4] Narayanaswami, C., Kamijoh, N., Raghunath, M., IBM’s Linux Watch: The Challenge of Miniaturization, IEEE Computer, Jan 2002, pp 33-41.

[5] C. Narayanaswami, M. Raghunath, N. Kamijoh, T. Inoue, “What Would You Do with a Hundred MIPS on Your Wrist?” IBM Research Report RC 22057 May 2001.