A version of this paper also appears in the *Proceedings of the Second International Symposium on Handheld and Ubiquitous Computing (HUC 2000)*

EVENTMANAGER: Support for the Peripheral Awareness of Events

Joseph F. McCarthy, Theodore D. Anagnost

Center for Strategic Technology Research (CSTaR)
Accenture
3773 Willow Road
Northbrook, IL 60062 USA

mccarthy@cstar.accenture.com

Abstract. EVENTMANAGER is a tool that supports peripheral awareness by enabling users to be notified when events of interest take place within their workplace environment. Our initial implementation of the tool allows users to specify events based on people and their locations within the physical environment, e.g., the event of Joe entering his office. We describe the context of the environment in which the tool is used, the event specification language, the features embodied in the interfaces and some potential extensions for future versions of the tool.

Keywords. Intelligent environments, ubiquitous computing, CSCW, human-computer interaction, social issues, peripheral awareness.

1 Introduction

A number of researchers have created tools designed to provide a peripheral awareness [Bly, *et al.*, 1993] of the activities of their colleagues. Some tools use video and/or audio information (using cameras, microphones, displays and speakers) – sometimes filtered – to provide awareness of the activities of other people [Dourish & Bly, 1992; Lee, *et al.,* 1997; Hudson & Smith, 1996]. Others have used more abstract or symbolic representations to provide this awareness [Ishii & Ulmer, 1997; Wisneski, *et al.,* 1998; Pedersen & Sokoler, 1997]. All of these tools rely on users' monitoring the displays or other representations – possibly at intervals – if they are interested in a particular activity or event.

EVENTMANAGER takes a different approach: rather than requiring users to continuously or repeatedly monitor "displays" that provide awareness information, we have built a tool that permits users to specify an event of interest involving people and locations. Users can then turn their attention to other tasks until they are notified that the event has occurred. For example, EVENTMANAGER enables users to be notified if or when:

- Joe arrives at work in the morning
- Eric and Jim are both in The Lab
- Ted returns to his office
- Anatole leaves the meeting
- The conference room is empty

If I need to ask Anatole a question, but he is now in a meeting, and I know he has another meeting immediately following this one, I can ask EVENTMANAGER to notify me when he is leaving one meeting and on his way to the next, presenting an opportunity to intercept him in the hallway. Without EVENTMANAGER, I would need to interrupt the current meeting, camp outside the meeting room until Anatole comes out, or wait until his meetings are over and he returns to his office to check for messages. Of course, if we had video cameras throughout the environment, I could monitor them to look for my opportunity, but this could require a great deal of attention over time.

In general, EVENTMANAGER provides opportunities to informally meet with one or more people [cf. Nakanashi, *et al.,* 1996] through an event specification language and interface through which I can specify one or more contexts in which I think such opportunities might exist. For example, if I need to ask Ted a question, I could ask EVENTMANAGER to notify me when he returns to his office, or if I need to talk with both Eric and Jim, I could ask EVENTMANAGER to notify me when they are both in The Lab. This approach is reminiscent of event-action specifications used for software-related objects and processes [Krishnamurthy & Rosenblum, 1995; Rosenblum & Wolf, 1997], but our tool applies to events that occur in the physical world rather than the digital world.

Peripheral awareness mechanisms can be characterized along two dimensions that define how much attention they require. One dimension is the attention *effort*, i.e., the amount of attention required each time the user wants to update his or her awareness. The other dimension is the number of attention *samples*, i.e., the number of times a user has to turn his or her attention to the awareness mechanism to gain or maintain awareness.

Most previous work in the area of peripheral awareness has focused on mechanisms that require low effort per sample, e.g., a quick glance at an awareness display (e.g., ACTIVEMAP [McCarthy & Meidel, 1999]), but may require multiple samples to monitor for specific events of interest. EVENTMANAGER, by contrast, requires a higher degree of effort to initially specify an event of interest, but then eliminates the need for subsequent attention sampling. While the low effort per sample mechanisms are clearly better for general awareness of activities in an environment, we believe EVENTMANAGER represents a better approach to providing awareness of specific, definable activities within an environment.

EVENTMANAGER works within the context of an intelligent environment [Coen, 1998], i.e., a physical space that can sense and respond appropriately to the people and activities taking place within it. One consequence of work in this area is a shift in perspective regarding human-computer interaction: rather than viewing people as *users* of computer systems, it becomes more appropriate to view people as *inhabitants* of computer-imbued spaces. In particular, EVENTMANAGER presumes that the environment can sense the locations of its inhabitants [Want, *et al.,* 1992; Harter & Hopper, 1994], and respond to the person who specified the event wherever he or she may be within the environment.

This paper will describe our EVENTMANAGER tool. We first describe the context within which EVENTMANAGER operates; we then present the EVENTMANAGER event specification language and its associated interfaces; we conclude with a discussion of some of our plans for extending the EVENTMANAGER.

2 Environmental Context

The environmental context in which we have designed and built the EVENTMANAGER is the physical space occupied by the Center for Strategic Technology Research (CSTaR®), a 16,000 square foot section of the second floor of Accenture Technology Park, in Northbrook, IL, USA. The CSTaR area includes 40 individual offices, four laboratories, two large conference rooms (the Group Discussion Lab, or GDL, and a VideoConference room), two small conference rooms, a break area with kitchenette and vending machines, three furnished open areas used for informal meetings and numerous hallways. There are approximately 30 members of the CSTaR group in Northbrook,[1] including researchers, programmers, technical writers and administrative staff.

We have installed an ArialView™ Awareness System [Arial Systems Corp.] within the CSTaR area, consisting of a network of over 70 ceiling-mounted nodes each housing an infrared sensor, radio frequency receiver and audio speaker, and a set of badges that transmit infrared identification signals every two seconds.[2] In addition to this hardware, the ArialView system includes components to process the signals and maintain badge location information in a Microsoft SQL Server 7.0 database, and a web browser interface for accessing and administering this information.

Some members of CSTaR have voiced privacy concerns about wearing a badge that allows them to be located in real-time or tracked over a period of time. Fortunately, we work in a profession in which our location does not reveal a great deal about our work (or play) activities: time spent in a colleague's office could represent an intensive exchange of project-related ideas, or a heated debate over whether a president committed impeachable offenses. One of the appealing features of a badge

[1] There are six members in another CSTaR group in Palo Alto, CA, USA, but their workspace is not yet incorporated into the environment(s) served by EVENTMANAGER.

[2] The ArialView system is similar in many respects to the Olivetti Active Badge System [Want, et al., 1992; Harter & Hopper, 1994], except that the current ArialView badges have a single two-position slider switch rather than two buttons, and the ArialView sensor nodes include RF receivers and speakers.

system is that anyone who objects to being located or tracked can simply not wear a badge; cameras and microphones are not so easy to avoid (though one can presumably at least control the devices installed in one's office).

We believe that most people are willing to relinquish some degree of privacy for what they perceive as a compensating benefit. For example, most people in the United States are willing to let grocery stores track their purchases via some kind of preferred shopper's card in exchange for small discounts received when they present the card to the cashier. It remains to be seen whether the members of CSTaR will perceive enough benefits from our suite of intelligent environment applications to warrant their continued wearing of badges. [3]

3 The EVENTMANAGER Tool

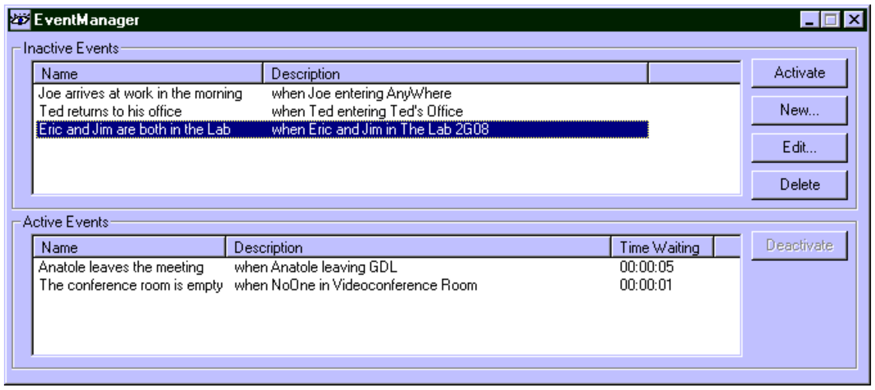
There are two interfaces for managing events in the EVENTMANAGER. The *event management interface* allows the user to create, modify, delete, activate or deactivate events.[4] The *event specification interface*, invoked from the event manager interface, allows the user to create or modify a single event, based on our event specification language. When an event's *conditions* are satisfied, the EVENTMANAGER has a number of *actions* it can use to inform the user. Each of these components is described in more detail below.

3.1 Event Management Interface

Figure 1 shows the primary interface to the EVENTMANAGER. The *inactive* events listed in the top frame have been specified by the user at some time, but are not currently being monitored by the system. The *active* events in the lower frame are specified events that are currently being monitored. Each event is assigned a *name* by the user, and has an associated *description* based on the primitives available in the event specification language. The active events shown in the lower frame also show how much time has elapsed since each event was activated.

[3] See Harper [1992], for a more thorough discussion of acceptance issues with respect to the use of badges in a research lab context.

[4] We will simplify our description of the tool by referring to *event specifications* as *events*, except in contexts where event specifications in the tool may be confused with events in the physical environment.



Inactive Events

Name	Description
Joe arrives at work in the morning	when Joe entering AnyWhere
Ted returns to his office	when Ted entering Ted's Office
Eric and Jim are both in the Lab	when Eric and Jim in The Lab 2G08

Active Events

Name	Description	Time Waiting
Anatole leaves the meeting	when Anatole leaving GDL	00:00:05
The conference room is empty	when NoOne in Videoconference Room	00:00:01

Figure 1: EVENTMANAGER's Primary Interface

There are four buttons on the right side of the primary interface:
- *New*: create a new event via the event specification interface
- *Edit*: view/modify an existing event via the event specification interface
- *Activate*: add the selected event(s) to the list of active events
- *Delete*: delete the selected event(s)

There is also a *Deactivate* button on the lower right side of the interface, used to deactivate an event before its conditions have been satisfied.

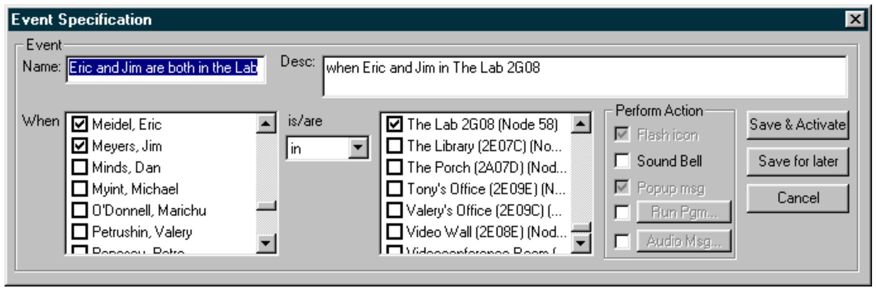
3.2 Event Specification Language and Interface

Figure 2 shows the Event Specification Interface to the EVENTMANAGER. The interface allows a user to specify any event within the constraints of our event specification language. Each event specification in our language has the general form:

when <*person*>+ is/are <*relationship*> <*location*>+ then <*action*>+

Where
- **<*person*>** is one or more members of the research group (the selection of multiple people is interpreted as a conjunction), or exactly one of the following special cases:
  - **SomeOne**
  - **NoOne**
- **<*relationship*>** is one of the following:
  - **entering**
  - **leaving**
  - **in**
  - **alone in**
  - **not in**

- ***<location>*** is one or more offices, conference rooms, open areas or hallways in the CSTaR area (the selection of multiple locations is interpreted as a disjunction), or exactly one of the following special cases:
  - **AnyWhere**
  - **NoWhere**
- **<action>** is one or more actions to take when the event occurs; this topic is covered in more detail below.



Event Specification

Event

Name: Eric and Jim are both in the Lab

Desc: when Eric and Jim in The Lab 2G08

When

- [✓] Meidel, Eric
- [✓] Meyers, Jim
- [ ] Minds, Dan
- [ ] Myint, Michael
- [ ] O'Donnell, Marichu
- [ ] Petrushin, Valery

is/are: in

- [✓] The Lab 2G08 (Node 58)
- [ ] The Library (2E07C) (No...
- [ ] The Porch (2A07D) (Nod...
- [ ] Tony's Office (2E09E) (N...
- [ ] Valery's Office (2E09C) (...
- [ ] Video Wall (2E08E) (Nod...

Perform Action

- [✓] Flash icon
- [ ] Sound Bell
- [✓] Popup msg
- [ ] Run Pgm...
- [ ] Audio Msg...

Figure 2: EVENTMANAGER's Event Specification Interface

Using this simple interface, we can specify the sample events listed in Section 1:

- when Joe is entering AnyWhere
- when Eric and Jim are in The Lab
- when Ted enters Ted's Office
- when Anatole is leaving The GDL
- when NoOne is in The GDL

The interpretation of a selection of two or more people as a conjunction (rather than a disjunction) was chosen because of our expectation that users would be more interested in knowing when two or more people are together in a location than in knowing when any one of a set of people is in a location. Another reason for this interpretation is that the latter class of events can be encoded by multiple single-person event specifications, e.g., "when Ted or Joe is in The Lab" can be represented by the two event specifications "when Ted is in The Lab" and "when Joe is in The Lab." Note that the interpretation of multiple selected people as a disjunction would not allow for alternate representations of conjunctions, i.e., we would not be able to specify the event of when Ted and Joe are both in The Lab at the same time.[5]

In addition to specifying one or more people, a single relationship, and one or more locations, the event specification interface requires users to *name* each event. There is also a *description* associated with each event, which is generated automatically from the selections made of the person(s), relationship and location(s). The name field was included to allow users to specify shorter identifiers for their events.

[5] The specification of two or more locations within a single event is interpreted as a disjunction because in our experience, people can not be in more than one location at any given time.

There are a variety of actions that the EVENTMANAGER can take once the conditions of an activated event specification have been satisfied:

- *Flash Icon*: EVENTMANAGER is part of a suite of awareness tools, collectively known as POCKETWATCH. When the main POCKETWATCH application is running, it has a special icon in the Windows 95/98/NT taskbar tray. This icon flashes whenever an event from the user's active event list takes place.
- *Sound Bell*: The computer's bell can be sounded to notify the user.
- *Pop-up Msg*: The user can ask for EVENTMANAGER to pop-up a small message-box window that lists the name of the event that has taken place.
- *Audio Msg*: The ArialView system provides the capability to send any arbitrary text message through a speech synthesizer and out to the speaker included in any of our ceiling-mounted nodes. One option for the user to be notified is to play a message that can be directed to wherever the user (or another inhabitant) is within the environment. This is a particularly useful option for notification since it provides users with an awareness of events of interest even when they are away from their offices.
- *Run Pgm*: We expect that a number of other people will want to extend the functionality of EVENTMANAGER in a variety of ways. One way we can permit some extensions early on, is to allow people to run any arbitrary application whenever an event takes place. We will discuss some of the ways that we plan to extend the EVENTMANAGER in the next section.

After creating or modifying an event specification, the user has three options:

- *Save & Activate*: adds the specified event to the list of active events
- *Save For Later*: adds the specified event to the list of inactive events
- *Cancel*: discards the specified event

4 Future Work

We currently envision a number of dimensions along which we want to extend the EVENTMANAGER. These include extensions to the event specification language, adding additional sensor information to augment the ArialView system, adding a synchronization capability for notifications, and adding new notification mechanisms.

4.1 Event Specification Language Extensions

We would like to enable users to specify a broader range of events, e.g., "when Eric is in The Lab, but Jim is not in The Lab," or "when three or more people are gathered together in any open area." One way we might do this is to specify ways to combine atomic events, such as those now enabled by the primary interface, using logical operators (AND, OR and NOT). Another potential direction is to allow variables and quantification, along the lines of predicate calculus representation.[6]

The current event specification language and interface only permit specification of discrete events. We would like to add the capability to include time intervals in the specification. This would enable us to specify events such as "when Anatole is alone in his office for more than 5 minutes," to distinguish this from when Anatole stops in his office to pick something up. We are also considering the addition of scheduling information to our event specifications so that a user can specify a time to activate and/or deactivate any event, or to create a regularly scheduled event, e.g., "when Joe arrives at work every day."

4.2 Integration with Other Information Resources

The current version of EVENTMANAGER relies entirely on information provided by our ArialView badge system. While this enables us to locate people throughout the environment, it doesn't tell us much about what they are doing. If we had access to information about whether someone was typing on their computer keyboard or whether their telephone was being used, we might be able to specify an event such as "when Anatole is alone in his office for more than 5 minutes and is not on the telephone."

It would also be useful to integrate calendar and scheduling information, particularly when someone is not locatable within the environment (due to a hidden badge, attendance at a meeting in a part of the building not covered by sensors, or being offsite). For example, if Anatole's schedule indicates that he will be in Palo Alto for three days, EventManager might notify a user who is specifying an event that involves Anatole returning to his office.

4.3 Notification Synchronization

The current implementation of EVENTMANAGER provides immediate notification when a specified event occurs. However, the environmental state can change before the notification has been observed. For example, I might be away from my office when a pop-up messagebox announces that "Anatole is available in his office." Upon returning to my office, I may see this message and then walk over to Anatole's office. Unfortunately, he may no longer be available.

[6] We should note that expanding the expressiveness of the specification language might yield undesirable consequences. If everyone specifies an event "when ten or more people are in The GDL" to await a critical mass of meeting attendees to arrive at our weekly seminar, we could experience deadlock, as everyone waits for everyone else to arrive. Of course, the seminar speaker might counter such a condition by specifying harassment events to send directed audio reminders to anyone not yet in the seminar room.

One possible solution is to incorporate a mechanism that automatically updates the pop-up message box when the state changes, e.g., "Anatole *was* available in his office." Another possible solution is to provide a capability to manually refresh the notification, e.g., a button that causes EVENTMANAGER to check the state of the system to confirm whether the condition indicated by the notification is still valid.

4.4 New Notification Mechanisms

Although we have included a catch-all notification mechanism in our current interface – the capability to run any arbitrary program when a specified event occurs – we'd like to investigate other mechanisms for notification to include in our default set. We are particularly interested in investigating the use of "tangible" notification mechanisms [Ishii & Ulmer, 1997; Wisneski, *et al.,* 1998] and other abstract or symbolic methods [Pedersen & Sokoler, 1997] for alerting users when events occur.

5 Conclusion

We have designed and built EVENTMANAGER, a tool that enables users to specify events of interest involving people and locations within an intelligent environment. The user is then notified if or when those events take place. We have considered a number of possible extensions, and are looking forward to feedback from our users to better understand how we might best expand the capabilities of this awareness tool.

References

1. Arial Systems Corp. ArialView™ Awareness System. http://www.arialsystems.com.
2. Sara A. Bly, Steve R. Harrison and Susan Irwin. 1993. Media Space: Bringing People Together in a Video, Audio, and Computing Environment. *Communications of the ACM,* 36(1), January 1993, pp. 28-47.
3. Michael Coen (Ed.). 1998. *Papers from the 1998 AAAI Spring Symposium on Intelligent Environments*. March 1998, Technical Report SS-98-02, AAAI Press.
4. Paul Dourish and Victoria Bellotti. 1992. Awareness and Coordination in Shared Workspaces. In *Proceedings of the ACM 1992 Conference on Computer Supported Cooperative Work (CSCW '92).* 330-337.
5. Paul Dourish and Sara Bly. 1992. Portholes: Supporting Awareness in Distributed Work Groups. In *Proceedings of the ACM 1992 Conference on Human Factors in Computer Systems (CHI '92).*
6. Richard H. R. Harper. 1992. Looking at Ourselves: An Examination of the Social Organisation of Two Research Laboratories. In *Proceedings of the ACM 1992 Conference on Computer Supported Cooperative Work (CSCW '92).* 330-337.
7. Andy Harter and Andy Hopper. 1994. A Distributed Location System for the Active Office. *IEEE Network* 8(1): 62-70.

8. Scott E. Hudson and Ian Smith. 1996. Techniques for Addressing Fundamental Privacy and Disruption Tradeoffs in Awareness Support Systems. In *Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work (CSCW '96).* 248-257.
9. Hiroshi Ishii and Brygg Ulmer. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In *Proceedings of the ACM 1997 Conference on Human Factors in Computer Systems (CHI '97).* 234-241.
10. Balachander Krishnamurthy and David S. Rosenblum. 1995. Yeast: A General-Purpose Event-Action System. *IEEE Transactions on Software Engineering,* 21(10), October 1995, pp. 845-857.
11. Alison Lee, Andreas Girgensohn and Kevin Schlueter. 1997. NYNEX Portholes: Initial User Reactions and Redesign Implications. In *Proceedings of the ACM 1997 International Conference on Supporting Group Work (GROUP '97)*.
12. Joseph F. McCarthy and Eric S. Meidel. 1999. ActiveMap: A Visualization Tool for Location Awareness to Support Informal Interactions. In Hans W. Gellersen (Ed*.) Handheld and Ubiquitous Computing*. Proceedings of the First International Symposium (HUC '99), Karlsruhe, Germany, September 1999. Lecture Notes in Computer Science, Vol. 1707. Springer – Verlag, Heidelberg. 158-170.
13. Hideyuki Nakanashi, Chikara Yoshida, Toshikazu Nishimura and Toru Ishida. 1996. FreeWalk: Supporting Casual Meetings in a Network. In *Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work (CSCW '96).* 308-314.
14. Elin Ronby Pedersen and Tomas Sokoler. AROMA: Abstract Representation of Presence Supporting Mutual Awareness. In *Proceedings of the ACM 1997 Conference on Human Factors in Computer Systems (CHI '97).* 234-241.
15. David S. Rosenblum and Alexander L. Wolf. 1997. A Design Framework for Internet-Scale Event Observation and Notification. In *Proceedings of the Sixth European Software Engineering Conference / ACM SIGSOFT Fifth Symposium on the Foundations of Software Engineering.*
16. Roy Want, Andy Hopper, Veronica Falcao, and Jonathon Gibbons. 1992. The Active Badge Location System. *ACM Transactions on Information Systems* 10(1): 91-102.
17. Mark Weiser and John Seeley Brown. 1997. The Coming Age of Calm Technology. In *Beyond Calculation: The Next Fifty Years of Computing*. Peter J. Denning and Robert M. Metcalfe, Eds. Springer Verlag. 75-85.
18. Craig Wisneski, Hiroshi Ishii, Andrew Dahley, Matt Gorbet, Scott Brave, Brygg Ulmer and Paul Yarin. 1998. Ambient Displays: Turning Architectural Space into an Interface between People and Information. In Norbert A. Streitz, Shin'ichi Konomi and Heinz-Jurgen Burkhardt (Eds.) *Cooperative Buildings - Integrating Information, Organization and Architecture.* Proceedings of the First International Workshop on Cooperative Buildings (CoBuild '98), Darmstadt, Germany (February 25-26, 1998). Lecture Notes in Computer Science, Vol. 1370. Springer - Verlag, Heidelberg. 22-32.