Learning to Detect User Activity and Availability from a Variety of Sensor Data

Martin Mühlenbrock, Oliver Brdiczka, Dave Snowdon, Jean-Luc Meunier
Xerox Research Centre Europe, Meylan, France
Martin.Muehlenbrock@acm.org

Abstract

Using a networked infrastructure of easily available sensors and context-processing components, we are developing applications for the support of workplace interactions. Notions of activity and availability are learned from labeled sensor data based on a Bayesian approach. The higher-level information on the users is then automatically derived from low-level sensor information in order to facilitate informal ad hoc communications between peer workers in an office environment.

1. Opportunistic meetings

In an office environment, unplanned interactions play an important role in the exchange of information as well as for the social cohesion of a group, their importance should not be neglected, as they are often instrumental to the organization [10]. In contrast to simply spontaneous encounters, opportunistic meetings occur where one person has a reason to talk with another but has not arranged to meet in advance and thus either seeks them out or waits until an encounter occurs by chance to raise the issue [9]. However, there is always the danger that the encounter might not take place (or not in a timely manner), leaving the issue unaddressed. In this paper we consider how such meetings, referred to here as IFFI's (for 'informal face-to-face interactions'), might be facilitated. In the first place by alleviating the burden of finding an appropriate time and place to meet while taking into account each person's constraints and activities. Facilitating IFFI's differs from making a rendezvous, since it enables a deep interleaving with current ongoing activities so as to support the setting-up of an interaction in near real-time. Secondly, these interactions should be supported in situ and in an ad-hoc manner, as they cannot be logistically pre-arranged. For instance, by discovering surrounding devices, such as printers or screens, that may support the interaction by providing convenient access to relevant documents. Finally, putting in perspective series of such interactions helps users to better articulate their work, for instance by reminding them of previous related interactions with the same persons or on the same topic.

As described above, these interactions are opportunistic in nature, and therefore depend upon the ability of the people involved to correctly interpret explicit and implicit cues. For instance, a colleague might be available to answer a short question if he has just finished a phone call and has not yet turned back to continue work on his presentation slides. On the other hand he might not be available if he has closed the door of his room. Hence the seizing of an opportunity for having an ad hoc face-to-face interaction is intertwined with the recognition of mostly visual attributes that go along with different activities in the working day. When workers are co-located in an office space, it is easy for them to grasp the corresponding signals. In settings where these are not perceivable, time may be wasted in seeking out opportunities for interaction, or people may indeed give up completely on the attempt, depending on the importance and the timeliness of the interaction.

Facilitating such meetings involves concepts of interruptibility and availability. Often interruptions are a mundane, normal part of work and it is through these interruptions that the work gets done [12], [13], [5]. However, there is also a considerable negative impact of interruptions, since interruptions can be difficult and disruptive. By developing a sensing infrastructure and contextual processing system we are aiming at providing helpful clues to people working in an office environment, when is a person interruptible, i.e. available for an opportunistic meeting.
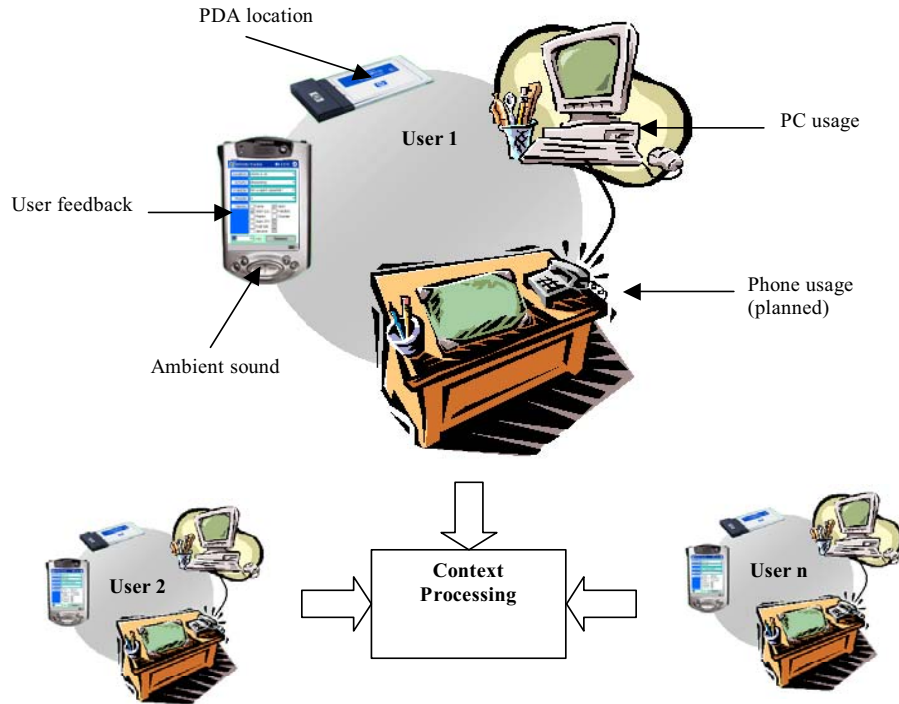
Figure 1: Sensing infrastructure

2. Sensing infrastructure

The application aims to reduce the burden of interruptions in the workplace, support informal workplace encounters by providing an indicator of a person's availability and support requests to find out when a particular person is available.

In order to do this we use a number of different sensing techniques in order to derive a person's activity (see also Figure 1)

- **PC usage:** Detection of users' keyboard and mouse activity on personal computers.
- **Phone usage:** Detection of phone usage (planned).
- **PDA location:** Determination of the location of users' PDAs.
- **PDA ambient sound:** Detection of ambient sound in the PDAs' vicinities.
- **PDA user feedback:** Explicit feedback on some context variables provided by the users.

Sensing the user activity on a personal computer is an important and easy source of information. Users' activity on personal computers is detected by a demon that runs on the PCs and notices typing and mouse movements. The most recent measurements are averaged, resulting in a smoothed increase and decline

of the usage value – this is done to prevent the system deciding that someone is not using their PC because they have paused for a couple of seconds to think. It seems obvious that a similar sensor on the PDA that detect writing on the screen and button presses would be useful. However, for performance reasons, the PDA's operating system (Windows CE) does not support hooks, as does the WIN32 API – hooks provide a simple means to implement the usage sensor and we have not found an equivalent in Windows CE. For the phone usage sensor, contact-less switches wired to a purpose built I/O card attached to a nearby PC's serial port allow the detection of a phone being taken off, and placed back on the hook.

The PDAs' locations are determined by using a wireless network. Wireless LANs are becoming more and more widespread, despite their current security weaknesses. A wireless LAN requires installing one or more access points, which act as a bridge between the wired and the wireless networks. With additional redundant access points, a location system can be obtained as a by-product of the wireless LAN, by triangulating the radio signal [2], [3]. Places are then identified by their radio characteristics in a calibration phase. Later on, a device can locate itself by measuring the current radio characteristics and comparing them with calibration data [1]. Both locating and tracking

are possible with this technique, depending if it is the device that computes its location or some server part of the infrastructure. Figure 2 shows the position of the network access points in a floor of our lab, where the accuracy is at the room level with a reliability of about 80%.
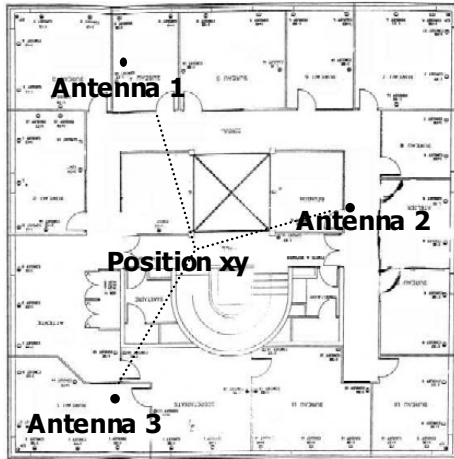

Figure 2: Wifi-based localization system

Furthermore, the PDAs are used to detect ambient sound in their vicinities. Each PDA is equipped with a microphone that is used to record sound samples of 10 seconds lengths. These sound samples are analyzed in comparison to a sample of the situation with the lowest sound level encountered so far. This sample defines a reference point for the no-ambient-sound situation.
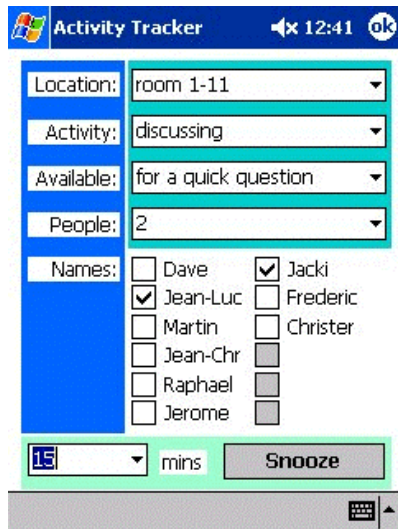

Figure 3: PDA interface for user feedback

In addition, a user interface has been developed for the PDA that prompts the user for information on his context in a regular fashion. This user information is used to label situations in order to create a set of training data. The user is asked to provide explicit feedback on a number of context variables. These include his location, the co-location with other people, which could be either people identified to the system or just the number of people present, activity and availability (see Figure 3).

In addition to the information given by the user, the user interface also records the time the user took to fill out the form. This piece of information allows for relating the information given by the user to the sensor data at the point in time just before the user interface popped up. This is important for two reasons. One the one hand, we assume that the user is likely to give information on the situation just before he was prompted for information, thus not saying that he is working on the PDA since at that point in time he is filling out the form. On the other hand, the filling out of the form itself produces sensor data such as ambient sound that is not related to actual user's situation, and hence it is necessary to be able to refer to the sensor data when the writing on the PDA's screen was not yet present.
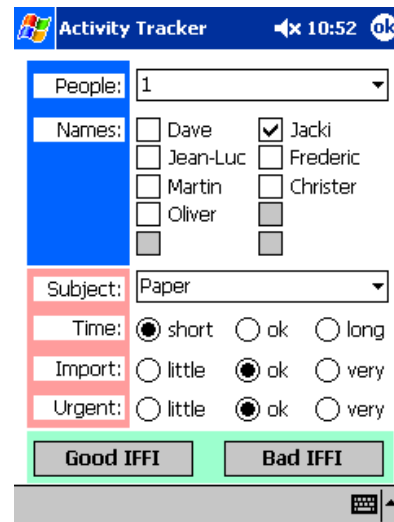

Figure 4: User feedback on an encounter

Furthermore, by using the hardware buttons, the user can also provide information to the system about encounters that take place. Three hardware buttons have been associated with functions for indicating the suitability or unsuitability of an encounter and end of an encounter, the latter allowing for an estimation of the encounters duration. Right after the meeting the user can input further information on the encounter concerning its timing, importance and urgency (see Figure 4). All this information is to improve the
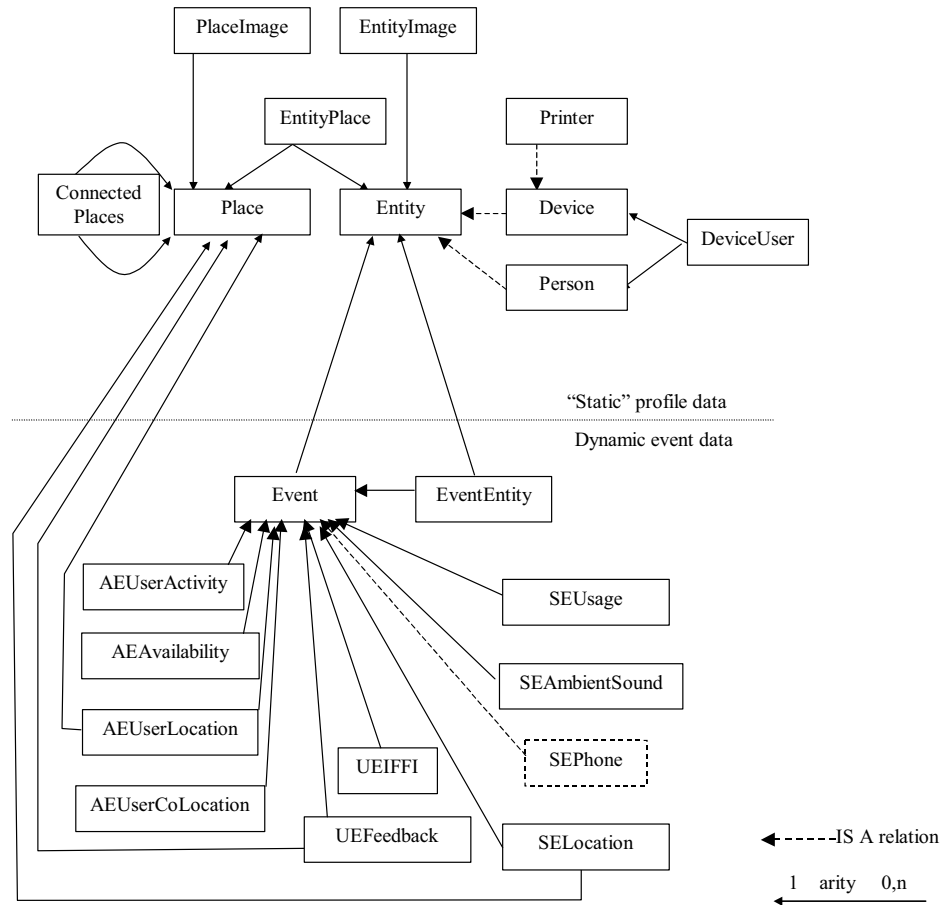
Figure 5: Database schema

systems ability to correctly estimate the user's availability for further encounters.

The various sensors send their information to a database residing on a server that can be accessed from both the wired and the wireless networks. The database contains "static" profile data as well as the dynamic event data (see Figure 5). The "static" profile data may vary over time (e.g. if someone is allocated a new PC or changes office) but comparatively slowly compared to the event data. The profile data names the entities (people and devices) and places that are referred to by the dynamic event data. Furthermore the profile establishes links between devices and places and people. For example the profile indicates that particular computers, PDAs and phones are associated with a particular user and that that user has his office in a particular place. It also indicates the normal function of places so that our software can find out if a

user is in a place that is someone's office or in a public space such as a meeting room or coffee area.

The tables associated with the dynamic event data store information about events generated by the sensors and events generated by higher-level components predicting activity and availability. Relations such as "EventEntity" link events to the entities (devices & people) related to those events. For both events and profile data we use a sort of inheritance scheme – for example the table Entity stores generic information about all entities and the table Person (a person is an entity) stores information specific to people that is not contained in Entity. Events are described by a unique id, the entity responsible for generating the event, a timestamp and a type. Each subtype of Event stores additional information such as a probability distribution of possible user locations, or the sound level measured by a PDA.
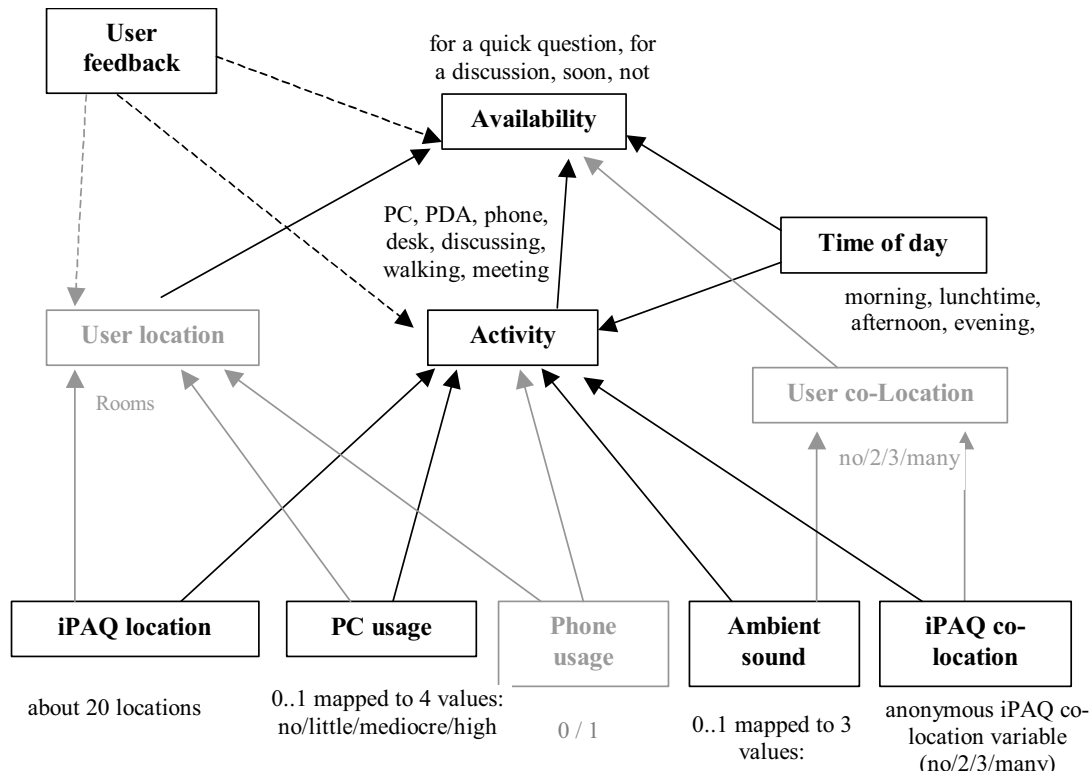
Figure 6: Learning dependencies (with planned parts grayed out)

3. Context processing and learning

The context processing consists of combining information from different sources and deriving an estimation of the users' situation. Of particular interest for the application are the activities and availabilities of the users. The set of relevant activities is comprised of

- **Single-person activities** like using a PC, using a PDA, and working on the desk,
- **Multi-person activities** such as phoning, discussing, or being in a meeting, and
- **Intermediate activities** like walking from one place to another, which result in a drastic change of context.

These activities are assumed to have a major influence on the level of a person's availability. Relevant classes of availabilities that are considered to be useful are being available for a quick question, being available for a longer discussion, being available soon, or not being available at all.

By using machine-learning methods the system is to find a connection between sensed information and situations as perceived by users, including also information on people's habits. This is in line with a growing interest in ubiquitous and contextual computing in deriving higher-level information from lower lever sensor data for the provision of some automatic services to the user in an unobtrusive way. We are focusing particularly on Bayesian methods, because they are well adapted to deal with the probability distribution in the localization data and have proven useful in many application domains. However, also the application of other methods such as decision trees or neural networks has been envisaged.

For our application the learning of classifications for activity and availability is of particular interest (see Figure 6). The user feedback on context variables in different situations provides higher-level information that allows for the labeling of lower level sensor information in order to use machine-learning techniques. The learned characterizations are then matched with new raw data to automatically detect types of activities and availability in situations. If the user does not agree on a certain characterization of his current situation, he can correct the description, thereby implicitly providing new training information for the machine-learning component.

3.1. Learning of activities

On the basis of labeled sensor data, we seek to learn probabilistic classifiers for relevant user activities, i.e.

- Activity $a \in ACT$ = {PC, PDA, phone, desk, discussing, walking, meeting}

We have as input for the machine learning sensor data and other basic information with the following ranges:

- PC usage: $PC \in$ {no, low, mediocre, high}
- Phone usage: $PH \in$ {yes, no}
- Ambient sound: $AS \in$ {no, low, high}
- Anonymous iPAQ co-location: $AC \in$ {no, 2, 3, many}
- Time of the day: $TD \in$ {morning, lunchtime, afternoon, evening, night}
- User's iPAQ location, given in form of a probability distribution $P_i(LOC)$ with $LOC \in$ {room₁, room₂, ..., roomₙ}

A Bayesian approach is used to determine the activity a_{MAP} with the maximum a posteriori (MAP) probability. As the location is given as a probability distribution, we have to sum over all rooms:

$$a_{MAP} = \arg \max_{a \in ACT} \sum_{LOC \in Rooms} P(a, LOC \mid PC, PH, AS, AC, TD)$$

By applying the Bayesian theorem (the constant denominator can be eliminated because of the argmax), we have:

$$a_{MAP} = \arg \max_{a \in ACT} \sum_{LOC \in Rooms} P(PC, PH, AS, AC, TD \mid a, LOC) P(a \mid LOC) P_i(LOC)$$

If we make the simplifying assumption that all sensor values are conditionally independent (Naïve Bayesian classifier), we have:

$$a_{MAP} = \arg \max_{a \in ACT} \prod_{S \in \{PC, PH, AS, AC, TD\}} P(S \mid a) \sum_{LOC \in Rooms} P(a \mid LOC) P_i(LOC)$$

In order to estimate prior probabilities for the Bayesian learning we count the number of occurrences of each activity in the user feedback with and without the respective sensor value detected and the sum of probabilities of room LOC in user feedback where activity was chosen. In order to get more reliable probability values, especially in the case of (nearly) empty user feedback, we used a simple LaPlace smoothing.

The user activity depends largely on the location where it takes place. We take that into account in weighting the a posteriori naïve Bayes probability with the probability distribution of the location. As we are working on all possible location values, we do not determine a discrete location for the activity yet. To do

so, we consider again the formula of the Naïve Bayesian classifier. As the term $\prod_{S \in \{PC, PH, AS, AC, TD\}} P(S \mid a)$ is constant for each activity, we can easily look for the maximal element of the sum $\sum_{LOC \in Rooms} P(a \mid LOC) P_i(LOC)$ once we have determined the argmax. This will give us the location that is the most probable for this activity with respect to the given location probability distribution.

3.2. Learning of availabilities

On the basis of the sensor data as well as derived data, we also seek to learn probabilistic classifiers for users' availabilities, i.e.

- Availability $av \in AVA$ = {for a quick question, for a discussion, soon, not at all}

We have as input for the machine learning higher-level information and other basic information with the following ranges:

- User's location given in form of a probability distribution $P(LOC)$ with $LOC \in$ { room₁, room₂, ..., roomₙ}
- User's activity given in form of a probability distribution $P(A)$ with $A \in ACT$ = {PC, PDA, phone, desk, discussing, walking, meeting}
- Time of the day: $TD \in$ {morning, lunchtime, afternoon, evening, night}
- User's anonymous co-location value $AUC \in$ {no, 2, 3, many}

A user's location and a user's anonymous co-location aren't basic sensor information, but are notions derived from sensor data. A user's location is inferred by relating the user's location habits (both sensed and from the user feedback) to iPAQ location and PC activity. A user's anonymous co-location combines ambient sound and anonymous iPAQ co-location.

A Bayesian reasoning determines the availability av_{MAP} with *the maximum a posteriori (MAP)* probability. As user location and activity are given as probability distributions, we have to sum over all activities and rooms:

$$av_{MAP} = \arg \max_{av \in AVA} \sum_{LOC} \sum_{A} P(av, LOC, A \mid AUC, TD)$$

By applying the Bayesian theorem, we have:

$$av_{MAP} = \arg\max_{av \in AVA} \sum_{LOC} \sum_{A} P(AUC, TD \mid av, LOC, A) P(av \mid LOC, A) P(LOC \mid A) P(A)$$

If we make the simplifying assumption that all sensor values are conditionally independent (Naïve Bayesian classifier) and we estimate that the calculated user location is independent from the calculated activity [$P(LOC|A)=P(LOC)$], we have:

$$av_{MAP} = \arg\max_{av \in AVA} P(AUC \mid av) P(TD \mid av) \sum_{LOC} \sum_{A} P(av \mid LOC) P(av \mid A) P(LOC) P(A)$$

As for the Bayesian learning of the activity, we estimate the prior probabilities by counting the number of availability in user feedback with and without the respective value of *AUC*, *TD* and the sum of probabilities of room *LOC* as well as the sum of probabilities of activity A in user feedback where availability was chosen. Again we smooth the values with simple LaPlace.

4. Results

In order to get some first results of the feasibility of the approach, we conducted several one-day experiments with different sets of users. We tested typical user situations like PC work, meeting, phoning etc. After having collected characteristic data during one day, we tried to classify new user-labeled situations the following day. Table 1 and Table 2 show the results of the activity and the availability detection in form of a confusion matrix. The training and test sets are comprised of 62 situations (day 1) and 27 situations (day 2), respectively. All situations included the activities "PC", "desk" or "discussing".

Table 1: Confusion matrix for activities

	PC	Desk	Discussing
PC	0.74	0.05	0.16
Desk	0.33	0.67	0.00
Discussing	0.00	0.00	1.00

The results of the detection of the activities "PC" and "discussing" are very good, because they rely directly on sensor information (PC activity, ambient sound). As the PC activity sensor smoothes its values, it does not immediately return to zero when the user stops working on PC and begins working on desk. That is why there is a quite high detection rate of the activity "PC", even though the user labeled it "desk".

The results of the detection of the availabilities "for a discussion" and "not at all" are excellent due to the fact that the users linked these availabilities especially to the time of day during the experiment. They didn't want to be contacted in the morning and they were available for a discussion in the afternoon. Furthermore, the users directly linked the activity "PC", which is very well detected, to the availability "soon".

Table 2: Confusion matrix for availability

	For a quick question	For a dis-cussion	Soon	Not at all
For a quick question	0.50	0.50	0.00	0.00
For a discussion	0.00	1.00	0.00	0.00
Soon	0.00	0.09	0.91	0.00
Not at all	0.00	0.00	0.00	1.00

In general, we found that a user's location is a strong indicator for his activity. This seems reasonable since in his room a user typically would be doing PC and desk work, whereas in his colleagues' rooms and meeting rooms he would usually be discussing or meeting, respectively. Nevertheless, we found that the current localization system is not sufficiently reliable. We have available an extended wireless localization system that, in addition to information from wifi connections, is using information from Bluetooth connections and infrared beacons, providing also a reliability measure that is based on the user's movement as detected by motion sensors attached to the PDA. The extended wireless localization system has not yet been used for the activity and availability learning, since it requires a considerable amount of training itself.

For the same reason, i.e. the requirement of larger amounts of training data, we did not cover mixed activities, i.e. situations in which the user performs two or more activities literally at the same time or in short alternations. However, these situations exist, e.g. a mixture of desk and PC activities, resulting in an arbitrary labeling when the user is prompted for specifying his activity in this moment of all situations. In some cases also the labeling of the users' availabilities may not be reliable, since it appears to be biased towards "available for a quick question", this may be due to the fact that at the moment providing availability information has no direct consequences for the users. This could change when the complete application is running as the effect of displaying this information to other users could result in a higher number of interruptions for quick questions, whereas the indication of "not available at all" would guarantee some degree of tranquility.

The smoothing of the PC usage information, i.e. the averaging of the most recent measurements, has shown to be problematic for the learning since it makes it difficult to relate the information to other sensors.
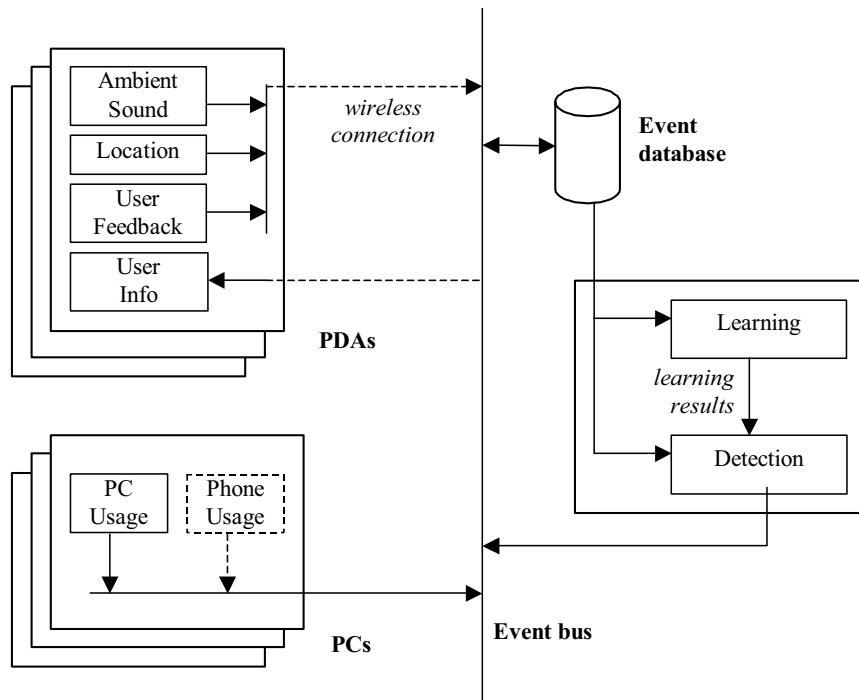
Figure 7: System architecture including flow of information back to the users

However, some smoothing seems to be beneficial as soon as derived information is shown to the user in order to prevent a "flickering" display of the situation description. Some further tuning of the system is required in order to find the optimum period of time to smooth mouse & keyboard detection over.

5. Providing the user with context information

The results of the learning of activity and availability notions are automatically included in a detection component, which is constantly monitoring the most recent events in the event database (see Figure 7). For each user the detection component derives an up-to-date context description based on the most reasonable situation estimation. The results of the automatic detection of situation features can be shown to the user in a prominent place on the PDA, e.g. as information item on an iPAQ's "Today" screen (Figure 8). The user may click on the information item to input further information whenever he finds that his context has been incorrectly guessed and a revision is relevant and necessary. In this way, the user is providing the system with new, labeled data, which can be used in a further learning round to further improve the situation detection.
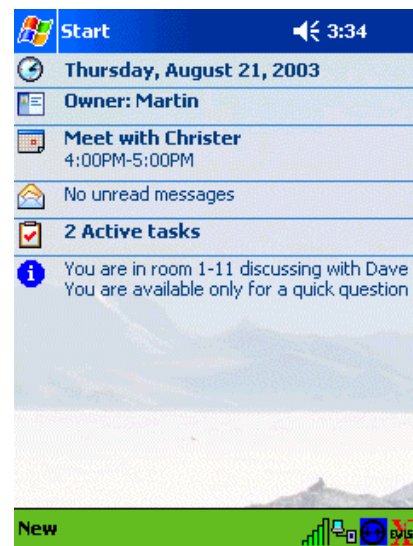

Figure 8: User context information on iPAQ's Today screen (see last item)

On the basis of the information generated by the context processing, the envisaged application then is to facilitate IFFIs. A user will create a request for an IFFI, or schedule its creation. The IFFI request may include the specification of one or more people and/or

of a topic, and possibly a deadline. The application will inform the user about what information is disclosed to other users, about the system guesses for location, activity, and availability, and about the IFFIs that are pending on the user. He also will be provided with peripheral awareness of the dynamics of the system such as the "birth/life/death" of opportunities and IFFIs. Furthermore, the system will provide a do-not-disturb mode. The agreement phase is optional, taking into account the cost and effort associated with an IFFI. Otherwise, also buddies' lists could be used. The system will suggest appropriate places by using profile information, and signal an immediate warning when a scheduled IFFI is rendered impossible.

6. Comparison to other work on contextual computing

Currently, there is a considerable interest in instant messaging systems, since they allow for quick and informal communications e.g. in the workplace; or elsewhere, if they are running on a mobile and connected devices. If these applications include context-sensitive features, they could be compared to the application we are developing, but in a distant communication scenario. ICQ is a well-known instant messaging system, which has a simple context sensor that automatically toggles the user status depending to his activity on the PC. This is in a way comparable to the PC activity sensor we are using in our infrastructure. Hubbub [8] is another system that provides an instant messaging system augmented with instant sound messages for mobile, networked devices. However, facilitation of face-to-face interactions has not been studied. In contrast, Proxy Lady [4] allows for the user to declare from the desktop PC the need to discuss an email with someone else. The PDA will then monitor the proximity of the other person and alert its user if they are close. However, this system only takes location into account, and does not include a richer model that is based on a variety of sensor information. In contrast to these systems that facilitate interactions, awareness systems simply provide visual information to their users and let them decide about using this information. Recently, applications for mobile users have been developed, e.g. Awarenex [13].

None of the systems mentioned so far, however, use machine-learning techniques to adapt the systems to different settings and users. However, the Microsoft Coordinate project [7] learns probabilistic models of peoples' meeting attendance and behavior based on information from PC use, organizational relationships,

time of day, and meeting attendance. It uses these models to decide whether messages are urgent and the best way to route them to the user; and how long it will be before the user returns to their office if he/she is away. However, this work relies on the provision of a labeled set of training data for a user. In addition the user is largely invisible to the system except when they are at their PC working. In our approach we are interested in covering a user's activity including situations in which people are out of their office, developing a more comprehensive tracking and analysis system. The Coordinate project focuses on predicting a user's presence at his work place (but not his availability), and at predicting a user's interruptibility when being in a meeting. In contrast to prediction, we are focusing on using a variety of easily available sensors for providing a rich context description of a user's current situation, including his activities and availability in all places. In addition, our approach is combining sensor information from several user, e.g. by using individual location information to derive co-location data.

In addition to Bayesian learning, other probabilistic methods have been used to learn and detect human activity. As a prominent example, an approach based on hierarchical hidden Markov models has been developed to learn the hierarchical structure of sequences of human actions [11]. The overall goal of this project is the extension of the functional capability of the elderly.

A group at Carnegie Mellon University is working on a technology [6] that uses cameras and microphones to detect when a person is busy. This currently relies on users being monitored over a period of time and manually indicating when they are busy so that the system can learn the important signs indicating that someone is busy. Although this is an important part of using ubiquitous computing to facilitate interaction it does not go far enough – we aim to detect both activities and whether the user is currently busy or not.

7. Summary and further work

We have presented an infrastructure for capturing contextual information in office environments to support and facilitate face-to-face interactions in an informal way. Sensors that are easily available in these environments are used to generate information on the users' situations. This information is abstracted to higher-level contextual data by using a Bayesian machine learning approach. The sensor infrastructure and context processing is general enough to be extended by other types of sensors. Sensors we have

IEEE
COMPUTER
SOCIETY

already worked on are location sensing via Bluetooth and infrared beacons as well as user movement sensing via motion sensors attached to PDAs. Further sensors that have been planned to be added that include personal agendas and profiles, room agendas, and image analysis of user activities from video camera input.

Acknowledgements

References

[1] Andreoli, J.-M., Castellani, S., Fernstrom, C., Grasso, A., Meunier, J.-L., Muehlenbrock, M., Ragnet, F., Roulland, F., & Snowdon, D. (2003). Augmenting offices with ubiquitous sensing. Proc. of Smart Objects Conference (SOC-2003), Grenoble, France, May.

[2] Bahl P. & Padmanabhan, V. N. (2000). Radar: An in-building RF-based user location and tracking system, In Proc. of the IEEE Infocom-2000, Tel-Aviv, Israel, vol. 2, Mar. 2000, pp. 775-784.

[3] Castro, P., Kremenek, T., Muntz, R. (2001). Nibble: A Component-based Location System, IEEE Computer (special issue on location systems).

[4] Dahlberga, P. & Sannebladb, J. (2000). The use of Bluetooth enabled PDAs: Some preliminary use experiences. In Proceedings of IRIS 23. University of Trollhättan, Uddevalla.

[5] Hudson, J., Christensen, J., Kellogg, W. A., & Erickson, T. (2002). I'd be overwhelmed, but it's just one more thing to do. Availability and interruption in research management. CHI (4) 1. 97-104

[6] Hudson, S.E, Fogarty, J., Atkeson, C.G., Avrahami, D., Forlizzi, J., Kiesler, S., Lee, J.C., and Yang, J. (2003). Predicting Human Interruptibility with Sensors: A Wizard of Oz Feasibility Study. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2003).

[7] Horvitz, E. Koch, P. Kadie, C.M., Jacobs, A. (2002) Coordinate: Probabilistic Forecasting of Presence and Availability, in Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence, Edmonton, Alberta, August, 2002.

[8] Isaacs, E., Walendowski, A., & Ranganathan, D. (2002). Hubbub: A sound-enhanced mobile instant messenger that supports awareness and opportunistic interactions. Proceedings of the Conference Computer-Human Interaction (CHI-2002), Minneapolis, MN, April.

[9] Isaacs, E., Whittaker, S., Frohlich, D., & O'Conaill, B. (1997) Informal communication re-examined: New functions for video in supporting opportunistic encounters. http://www.izix.com/pro/lightweight/video.php

[10] Kraut, R., Fish, R., Root, B., & Chalfonte, B. (1990). Informal communication in organizations: Form, function and technology. In S. Oskamp & S. Spacapan, (Eds.), People's reactions to technology in factories, offices and aerospace, The Claremont Symposium on Applied Social Psychology, Sage Publications, 145-199.

[11] Lühr, S., Bui, H.H., Venkatesh, S., West, G.A.W. (2003) Recognition of Human Activity through Hierarchical Stochastic Learning. In Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom-03): 416-421. Fort Worth, Texas, USA

[12] Martin, D., Rouncefield, M. & Sommerville, I. (2002). Applying patterns of co-operative interaction to work (re)design: e-government and planning. CHI letters 2002. 4 (1), 235-243

[13] Rouncefield, M., Viller, S., Hughes, J. A., Rodden, T. (1995). Working with 'constant interruption': CSCW and the small office. The Information Society v11, n3: 173-188.

[14] Tang, J.C., Yankelovich, N., Begole, J., Van Kleek, M., Li, F., & Bhalodia, J. (2001). ConNexus to Awarenex: Extending awareness to mobile users. In CHI Letters: Human Factors in Computing Systems. 3(1): pp 221-228.