The Calendar as a Sensor: Analysis and Improvement Using Data Fusion with Social Networks and Location

Tom Lovett, Eamonn O'Neill
University of Bath
Bath, UK, BA2 7AY
tom.lovett@vodafone.com
eamonn@cs.bath.ac.uk

James Irwin, David Pollington
Vodafone Group R&D
The Connection, Newbury, UK, RG14 2FN
james.irwin@vodafone.com
david.pollington@vodafone.com

ABSTRACT

The shared online calendar is the de facto standard for event organisation and management in the modern office environment. It is also a potentially valuable source of context, provided the calendar event data represent an accurate account of 'real-world' events. However, as we show through a field study, the calendar does not represent reality well as genuine events are hidden by a multitude of reminders and 'placeholders', i.e. events that appear in the calendar but do not occur. We show that the calendar's representation of real events can be significantly improved through data fusion with other sources of context, namely social network and location data. Finally, we discuss some of the issues raised during our field study, their significance and how performance could be farther improved.

Author Keywords

Calendar, data fusion, meeting, event, context, context awareness, social network, contacts, location.

ACM Classification Keywords

H.5.3 Group and Organization Interfaces: Collaborative Computing, H.4.3 Communications Applications: Information Browsers, H.5.2 Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms

Design, Experimentation, Human Factors.

INTRODUCTION

The shared calendar has long been an effective tool for collaborative organisation and management, especially within enterprise settings. Not only is it a useful indicator of presence and availability but many people use it for purposes such as archiving [11] and content management [2]. Online calendar events are a potentially useful source of context, as they typically contain data on future presence that may otherwise be unavailable or unobtainable using traditional sensors.

However, the online calendar alone is limited as a 'sensor' for a number of reasons. First, it is unlikely to be a consistently accurate representation of the real world due to events not occurring, or its common use as a to-do list. Also, events may occur outside their allotted time window, and actual event attenders may differ from those invited. If a system were to use the calendar as a virtual sensor, it would ideally require as little deviation from the real events as possible. Data archiving or mining systems using the calendar for indexing could experience an impact on reliability for the same reasons. Secondly, the calendar does not provide dependable real-time information. For example, within the Microsoft Communicator application, a user's availability is automatically changed to 'in a meeting' when a calendar event occurs. If the user is planning to attend the meeting late, or has left the meeting early, it is not reflected in her online presence. Thirdly, reminders and to-do list items are also commonly registered as events on such systems and again the user's availability will be listed as 'in a meeting' when in reality she is not.

In this paper, we make the following contributions. First, we compare an enterprise online calendar to real world events through a field study, showing that genuine events make up only a small fraction of the total calendar events. For the genuine events, we show variations in time when compared with their calendar entries, differences between invited attenders and actual attenders, and inconsistent location specification in calendar entries.

Secondly, we present two heuristic methods of data fusion that combine the calendar with social network and location data to produce a real-time multi-sensor interpretation of the real-world events. We apply these methods to the data gathered in the field study, showing that the calendar can be significantly improved as a sensor and indexer of real world events through data fusion. Consequently, useful contextual data within the calendar can be uncovered, enabling the development of new applications or improvements to existing applications that make use of presence and availability.

BACKGROUND AND RELATED WORK

Context awareness has generated substantial interest in both the academic and industrial research communities. Here we briefly review related research that has studied or made use of calendar systems to derive context and recent work investigating the concept of event detection.

Mobile Context Awareness

Mobile device based context aware systems and applications are active research areas. Early work by Schmidt et al [13] used physical and logical sensors on a mobile device to demonstrate situational awareness with a similar layered model approach to ours (see Figure 2). Indulska and Sutton [5] discuss the idea of physical, virtual and logical sensors when applied to location management in pervasive systems. Of particular relevance here is the authors' layered framework, which features a fusion layer combining abstracted outputs from the different sensor types to enhance location information given by each of the standalone sensors. Fogarty et al [1] present a context aware communication client that uses data fusion to provide a better interpretation of how interruptible a user may appear to her colleagues.

The Shared Office Calendar

Forecasting activity, presence and availability through the use of calendars are popular research topics. Various projects have investigated the usefulness of the calendar in coordination and collaboration [8]; how the calendar is used [11] and applications to augment the shared personal calendar [15]. Mynatt and Tullio have contributed a number of studies on the use of the calendar and its applicability to future availability. In [7] they present the calendar as a sensor that provides a likelihood of future presence and location. Their application, Ambush, uses a Bayesian model to predict attendance likelihood for calendar events based on previous attendance. In this work, they also show that co-workers in enterprises use their shared calendar to 'ambush' colleagues for *ad hoc* meetings when they are not busy. In [14] they discuss the deployment of the application and implications of using forecasting in groupware system design.

In [16], and this is perhaps the most relevant to our work, Tullio states that during his studies, events were attended between 52% and 63% of the time. Citing an unpublished study by Bradner, he notes that calendars are often cluttered with events that were not attended, highlighting his desire to provide a more informed interpretation of users' schedules. Further uses of the calendar in enterprises have been studied by Palen and Grudin [10]. They show that office workers frequently use shared calendars to infer the presence and availability of their colleagues.

Other related work investigating the shared office calendar was undertaken by Horvitz et al [4], whose Coordinate system uses machine-learning predictive modelling to forecast user presence and availability.

Event Identification

Research has also been conducted on the concept and definition of events, as well as their identification through various sensory inputs. Westermann and Jain [18] present a common model to describe the facets of an event, broadly classified around key areas of context, i.e. temporal, spatial and social. Event detection is discussed at length by Xie et al [19]. In this work, the authors investigate and classify various event detection systems and their uses in problems such as media management and data mining. They draw on the 5W1H model of event classification which is similar to the model presented by Westermann and Jain. They also look at the role of context when detecting events, alluding to the advantage of *a priori* knowledge, or planning, in event classification. The event detection analysed by this work is generally undertaken *post hoc*, i.e. mining post-event multimedia in the attempt to detect the event itself. In contrast, our work focuses on the *real-time* aspects of event detection, detecting events as and when they happen.

Eagle and Pentland's BlueAware and BlueDar systems [12] use similar methods to ours when identifying co-present system users. They fuse user profile data with this co-presence information in an attempt to induce 'social serendipity' between proximate users who do not know each other. Real-time meeting detection is investigated by Wang et al [17], who present a meeting identification system using sensor-fusion. They attempt to measure meeting start and end times through pressure sensors in seats, with a 95% success rate. Other research into the importance of meeting semantics, knowledge of meetings and capture of meeting metadata is discussed in [3]. This work suggests that there is value in the consistent and accurate semantic capture of meetings and the advantages these data bring to knowledge management and legacy searching problems.

THE CALENDAR VS REALITY

We began with a 6-week field study of a workplace (approx. 200 employees) within a large enterprise. The key employee roles focus on software development and engineering. Scheduled meetings are commonplace among the employees. The environment was representative of many modern offices, although we appreciate the impact on generalisability of studying a single environment.

We compared the workplace online calendar events to their equivalent 'real-world' events in order to evaluate the performance of the calendar as a virtual sensor, and recorded additional user location and social context data with which we aimed to investigate improving the calendar's performance as a virtual sensor through data fusion.

Method

The enterprise studied uses the Microsoft Outlook application as a shared calendar tool. We recruited 20 participants from two closely related teams, with 11 and 9 participants from each team respectively. Two participants had managerial roles while the rest were software developers.
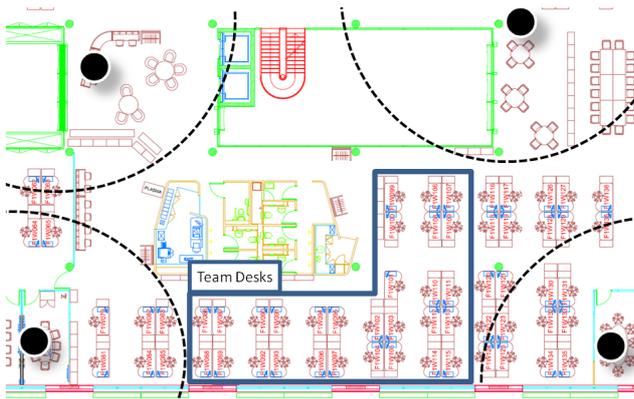
Figure 1. A plan of the office used in the field study, showing the location and approximate measured coverage of the four static devices and team desk area.

There were two key data sets in our comparison: the complete set of online calendar events for our participants over the 6-week period, and a record of actual events involving our participants for the same period. The calendar facility offered by Outlook is commonly used in practice, as we see below, in two very different ways: i) to create appointments, typically with other Outlook users, and ii) to create personal reminders. To Outlook, however, both appear as the same 'event' object.

Calendar Events: In order to capture the calendar events, we obtained programmatic access to each participant's Outlook application throughout the duration of the study. Calendar events were captured 'live', i.e. we recorded the entries in real-time, storing any changes made by the participants during the study, such as amended invite lists, times, locations and event names. The Exchange Server assigns each event object a unique ID, so every event had a single identifier even if it were stored in multiple calendars. Events such as private appointments that were not accessible through the shared system were ignored.

Real-World Events: Real-world events represent what actually happened in terms of meetings involving two or more of our participants. Our record of these was obtained through a combination of three methods: ethnographic observation, participant interviews and participant diaries. In the ethnographic study, we observed the participants during their working days and recorded any events they were involved in. However, we could not monitor all the participants all of the time, so we instructed them to keep an event diary for the 6-week period in which they recorded details of each of their events. Finally, we conducted weekly interviews with participants. This included examining their diaries for the week, validating our ethnographic data – i.e. the events we had recorded – against their diary entries and verifying our recorded events with them through questioning and discussion.

We collected additional contextual data on our participants, including the following:

Location: Each participant was given a mobile device running the Windows Mobile operating system. We built a small application to run on the devices that performed a Bluetooth scan of the surrounding environment at 2 minute intervals. After each scan had completed, the application uploaded the timestamped results to our server using either 802.11 Wi-Fi or a GPRS mobile data connection. In order to estimate participant location, we placed 4 static devices in known positions within the workplace. These locations, shown in Figure 1, were the key meeting areas for the two teams, and the static devices served as identifiers for each location. Each device performed a Bluetooth scan at 1 minute intervals and uploaded the timestamped results to our server. Thus, if a participant were to move within the 'hotspots' in Figure 1 there would be a good chance, subject to the usual vagaries of Bluetooth scanning [9], of their

Category	Description	Symbol	No. of events (% contribution)
Genuine Event	A shared online calendar event involving one or more study participants that represents a genuine real-world event.	G	38 (8%)
Placeholder Event	A shared online calendar event involving one or more study participants that does not represent a genuine real-world event because the calendar event was created and did not occur, e.g. a recurring daily meeting that did not occur on a particular day.	P	152 (32%)
Personal Reminder	An online calendar 'event' created by a participant simply as a reminder to herself, e.g. 'Backup Files', and without inviting any other 'attenders'.	R	232 (49%)
Shared Reminder	A shared online calendar event created as a reminder to two or more study participants, with 'attenders' 'invited' only to enable the sharing.	S	52 (11%)
Out of Scope	A shared online calendar event that i) involves a single study participant and other users not involved in the study or; ii) was external to our meeting areas e.g. at a different site or; iii) is outside office hours.	Z	120 (n/a)

Table 1. Calendar event categorisation and the contribution of each category to the overall calendar.

mobile device reporting a Bluetooth sighting of a static device and vice versa. The area in which the team desks are located was not covered by a static device. This was to minimise the risk of the static devices interfering with each other or reporting ambiguous results due to participants being sighted in two hotspots at once. Although ambiguity was thereby minimised, this decision had consequences for system latency that are discussed later in the paper.

Contacts: In addition to accessing participants' calendars, we also captured the manual contacts, i.e. non-corporate address book contacts, of each participant through their Microsoft Outlook application. This was also recorded 'live'; i.e. when contacts were added, changed or deleted the action was communicated to our server. Existing contacts, i.e. contacts added to the address book before the study, were also captured when the applications were installed on the participants' computers.

Results

By the end of the field study, we had collected 594 unique online calendar events from the participants. In contrast, we recorded only 38 distinct real-world events involving two or more participants, each of which corresponded to one of these calendar events. In Table 1, we categorise the calendar events according to various observed characteristics. Events in set Z were beyond the scope of our analysis since we were studying only a subset of users from the whole enterprise, in a sample location and during normal working hours. Excluding Z provided more representative sample calendar data for our participants. Removing this out of scope subset from the set of 594 calendar events leaves 474 in scope events for us to consider. Table 1 also lists the percentage contribution of each category, *excluding* the events in Z. We define the complete set of events within our analytical scope here as C where:

$$C = G \cup P \cup R \cup S$$

In Table 1, we can see that nearly half the events in the study are actually personal reminders R. The set of placeholders P(152) accounts for a third and the set of genuine meetings G(38) accounts for only 8%, outweighed even by the set of shared reminders S. A simple exclusion rule can be applied to distinguish the personal reminders R from the other categories: ignore all events with fewer than two invited attenders (including the calendar event creator). However, it is not so trivial to differentiate between a genuine event G(38), a placeholder event P(152) and a *shared* reminder S(52) as they are all in exactly the same format in the online calendar and all have two or more invited attenders.

For later comparison, we class G(38) as 'successful event identifications' and the union of P(152) and S(52) as 'false event identifications'. For the 38 genuine events, the calendar also contains 204 false events - a large figure that will affect the calendar's performance as a sensor. Next we compare the characteristics of the 38 recorded real-world events against their equivalent calendar events G(38) in order

to quantify the similarity between them. To compare locations we matched, by observation, the location field of the calendar events with the actual location of the recorded real-world events. To compare the attenders, we used the Jaccard index of set similarity, $J(V_r, V_c)$, where V_r is the set of recorded real-world attenders and V_c is the set of invited calendar attenders. The mean Jaccard index, where N is the number of recorded real-world events, is given by:

$$\overline{J(V_r, V_c)} = \frac{1}{N} \sum_{j=1}^{N} \left[\frac{|V_r \cap V_c|}{|V_r \cup V_c|} \right]_j$$

A mean Jaccard index of less than 1 shows that the calendar has either listed an attender who was not part of the recorded real-world event, or has failed to list an attender who *was* part of the recorded real-world event. For clarity, we break these down as follows: c represents a correct attender identification in G(38), f a false identification, and a a failed identification. Table 2 lists the results of the comparison, from which we can draw various insights.

The time differences between the calendar and recorded real-world events illustrate the phenomenon behind the availability problem described earlier, where a system such as Microsoft Communicator will list a user's presence as 'in a meeting' when in reality she is not. As the results show, the majority of recorded real-world events in this enterprise start *later* than indicated by the calendar, and the standard deviation figures show a large variability between calendar and recorded real-world event start and end times. The low location similarity of 0.11 is a result of the calendar event location field not being consistently populated, illustrating that the calendar data does not provide a good record of event location. A high mean Jaccard index shows the calendar to be a good indicator of real-world event attendance, as is also shown by the total number of correct attender identifications

Comparison Metric	Calendar vs. Real-World
Start time (μ_s, σ_s) (nearest 5 min)	(-25, 25)
End time (μ_e, σ_e) (nearest 5 min)	(-5, 15)
Location	0.11
$\sum c$	113
$\sum f$	16
$\sum a$	9
$\overline{J(V_r, V_c)}$	0.89

Table 2. Calendar events vs real world events: μ = mean, σ = standard deviation. Negative μ indicates the calendar time is earlier than the equivalent real-world event time

(113/122). However, this was revealed through *a posteriori* knowledge of the recorded events obtained through our ethnographic observations and participant diaries. If we were to look at the calendar events alone without the *a posteriori* data, it would be very difficult to distinguish the genuine events $G(38)$ from the placeholder events $P(152)$ and shared reminders S. The calendar alone, therefore, is not a good representation of reality; useful contextual data contained within $G(38)$ is effectively hidden by $P(152)$ and S.

To summarise, we found that in a typical enterprise calendar, the vast majority of 'events' were reminders or placeholders, and few were actually representations of genuine real-world events. We also found that the similarity between recorded real-world events and their calendar equivalents is variable, and that the calendar is not a reliable indicator of real-world event times or locations. It is, however, a reasonable indicator of attendance but this information is hidden among the false events and not easily discernable without *a posteriori* knowledge. Thus, without additional exogenous knowledge of the context, it is difficult to differentiate between genuine events, placeholders and shared reminders, making the calendar alone an unreliable virtual sensor.

DATA FUSION: HYPOTHESIS AND DESIGN

Following on from our analysis of the calendar, we formulated the following hypothesis:

The shared calendar is a potentially valuable source of context but its usefulness as a virtual sensor is limited if its content is not a good representation of reality. The similarity between calendar event data and reality can be improved through fusion of the calendar with other sources of context.

To achieve this, we must minimise the cardinality of sets $P(152)$ and $S(52)$ while simultaneously minimising the impact on the cardinality of set $G(38)$. In other words, our objective is to make $C(474)$ as similar to $G(38)$ as possible. We also aim to improve the metrics of the calendar events listed in Table 2. To do this, we developed a meeting detector system to focus on data fusion using sources that represent the key aspects of a real-world event: the temporal (start and end times), the spatial (location and co-presence) and the social (attenders) [6, 18].

Our Design Model

In our system design, we used the simple layer model shown in Figure 2. Using a bottom-up approach, we modelled physical sensors, e.g. GPS, and virtual sensors, e.g. the calendar, in a data 'Gathering' layer. The next layer up is an 'Aggregation' layer. It consists of enabler modules that receive inputs from the underlying sensors. The function of each enabler is to apply low-level fusion across its sensory inputs, providing a best estimate for output to the 'Processing' layer. For event identification, we used the following enablers:
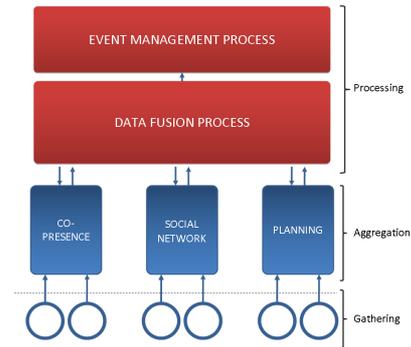

Figure 2. Layer model showing enablers and processes

Co-presence: We make an initial assumption that physical meeting events must involve two or more proximate users. To determine a set of proximate users, we require data from time and location sensors. This enabler samples from these sensors to return a set of proximate – or co-present – users based on specified location and range inputs.

Social Network: We also assume that event attenders have some form of interrelationship, e.g., email contact, in order to organise the event itself. To determine this, we require data from sources that represent interrelationships, so this enabler samples from available social network sensors, e.g. mobile device contacts lists, to output a social graph for a specified user or set of users.

Planning: We assume events have some form of planning. This can range from a simple *ad hoc* request, e.g. "Do you have time for a quick chat?" to a formal calendar appointment. This enabler samples data from available planning sensors – including the calendar itself – in order to output an initial set of inferred events for a specified user or set of users at a given time.

Next, we consider higher level data fusion of the enabler outputs in the Processing layer of Figure 2. To do this, we designed two processes – the Data Fusion Process and the Event Management Process.

Combination No.	Enabler Output 1	Enabler Output 2	Enabler Output 3
1	CP	SN	PL
2	CP	P	SN
3	PL	SN	CP
4	PL	CP	SN
5	SN	CP	PL
6	SN	PL	CP

Table 3. The six candidate data fusion methods using outputs from the three enablers. CP = Co-presence, SN = Social Network and PL = Planning.
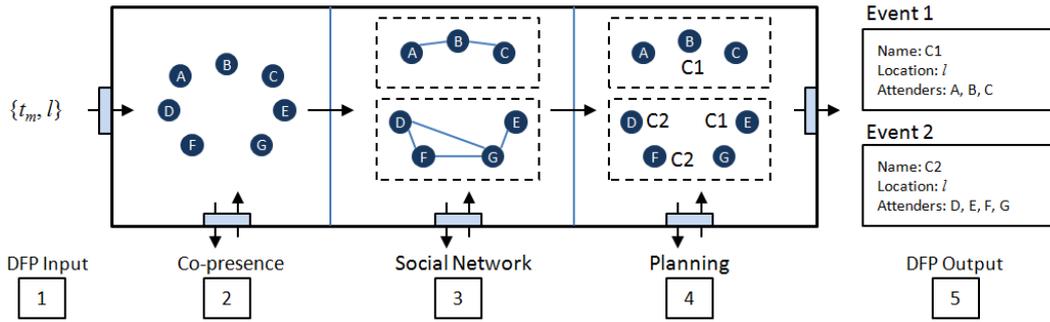
Figure 3. Data fusion method 1; illustrating how the method uses the enablers to group co-present users into subgraphs using social ties, before analysing planning data within each subgroup to create events. DFP = Data Fusion Process

The Data Fusion Process

Here we derive two methods of data fusion based on combinations of the enabler outputs described in the previous section. It is important to note that these methods are *heuristic*: due to their dependence on the availability and quality of the sensor data, they do not guarantee that an event will be detected, or detected correctly, every time. We first analysed all possible combinations of the enabler outputs listed in Table 3, where each combination represents a possible data fusion method. To elaborate: each fusion method begins with the output from Enabler 1 which cascades through Enabler 2 and, subsequently, Enabler 3.

We chose to start the Data Fusion Process using the Co-presence enabler, due to the real-time aspect of event detection. If we were to commence the Data Fusion Process using the Social Network enabler – combinations 5 and 6 in Table 3 – then we would require knowledge of a potentially vast and temporally dynamic social network in which changes to the network, e.g. creation of new edges, vertices or clusters, could be detected as evidence of an event occurring. However, this approach is unlikely to be able to provide *consistent* indicators of events so we dismissed combinations 5 and 6. We then considered 3 and 4. Beginning the Data Fusion Process with the Planning enabler would be similar to using the calendar events as triggers, i.e. the process would begin at the start time of a calendar event.

As we have seen, however, the calendar alone is not a good indicator of context and using Planning as the first enabler would subsequently affect the performance of the succeeding enablers. Thus, we dismissed 3 and 4. This leaves 1 and 2, starting with the Co-presence enabler. If we follow our assumption that physical meeting events require at least two proximate users, then using co-presence as an initial indicator of an event occurring is appropriate. Combinations 1 and 2 were therefore implemented as our data fusion methods.

As both data fusion methods begin with the same enabler, the key difference between the two is the prioritisation of the remaining two enablers. We had no evidence on which to predict whether grouping co-present users by social ties before planning ties (method 1) would perform better than grouping by planning ties before social ties (method 2), so we implemented and analysed the performance of both. For brevity, we explain the operation of the Co-presence enabler once, as it will output the same results for both methods. Next, we describe the path for method 1 and its outputs before describing the equivalent for method 2. We refer to Figure 3 for method 1 and Figure 4 for method 2, as they provide visual examples of the operation and differences between the methods when executed using the same dataset. In both Figures 3 and 4, Stage 1 begins with a request to the Co-presence enabler for a set of users U_c who are co-present in a particular location l at time t_m, specified in the request. If we assume U to be a set of users u who can each provide a
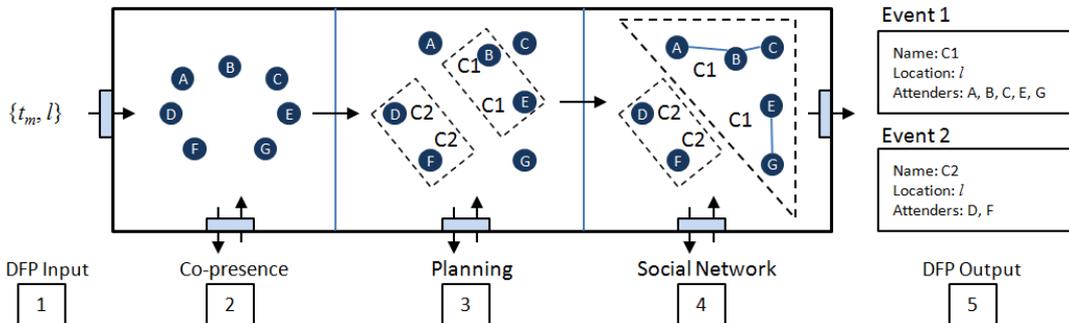

Figure 4. Data fusion method 2; illustrating how the method uses the enablers to group co-present users into subgraphs using planning ties, using remaining users social ties to extend the groupings. DFP = Data Fusion Process.

location measurement l_u with timestamp t_u then at time t_m, U_c is formally defined as:

$$U_c = \{u : u \in U \wedge |l - l_u| \le p \wedge (t_m - t_u) \le t\}$$

where $P(152)$ is a specified proximity threshold or range, and t is a specified time window in which to consider a user's location update as 'live'. U_c is returned at Stage 2 in both Figures 3 and 4. Following our assumption that an event must contain two or more attenders, we stop execution if the cardinality of U_c is less than two.

Method 1 path: At Stage 3 in Figure 3, method 1 passes U_c to the Social Network enabler, which searches for social ties between the users and outputs a collection of subgraphs in the format $G(V, E)$, such that $V \subseteq U_c$ and E is a set of *social* ties between the users in V. If the cardinality of V in any subgraph is less than two then, following our critical assumption, G is removed from the collection of subgraphs. Two subgraphs have been returned from the Social Network enabler illustrated in Figure 3.

At Stage 4, the collection of subgraphs is passed to the Planning enabler, which searches the planning data for users in each subgraph at time t_m, outputting an n-best ranked list of calendar entries per subgraph. If there is an unequivocal best ranked calendar entry, the method will create an event with the name of this calendar entry and attenders V. We assume V_{cal} to be the subset of V such that the each member of V_{cal} is a calendar entry owner. If there is a tie in the calendar ranking output, the calendar entry of the V_{cal} member with the highest degree centrality, i.e. the most ties to other nodes in the subgraph V, is chosen. If this too is ambiguous, one of the entries is selected at random. There is a risk here that the method will output a false event or fail to identify a genuine event, but this is a last resort in the case where further evidence to decide a tie is not available.

If a calendar entry cannot be found for any particular subgraph, we decide that there is not enough evidence to support event identification and the subgraph is discarded. It is worth noting that, although this action will disregard a good number of false event identifications, it will also ignore all cases of *ad hoc* events, i.e. events not linked to any calendar entry, such as an unplanned meeting or the legendary 'water-cooler chat'. *Ad hoc* events are out of scope for this analysis, but their identification is an interesting research challenge.

Method 2 path: As illustrated in Figure 4, at Stage 3 U_c is passed to the Planning enabler that searches for any planning data at time t_m for the users in U_c. The method subsequently creates a collection of subgraphs in the format $G(V, E)$, where $V \subseteq U_c$ and E is a set of *planning* ties between the users in V, i.e. the members of V all have the same shared event in their calendars. Users with no calendar entry for time t_m are not grouped. Time is specific here but there is an argument, discussed later, for making it fuzzy. At Stage 4, the collection of subgraphs and the ungrouped users are

passed to the Social Network enabler, which attempts to link the ungrouped users to the subgraphs based on social ties. An ungrouped user is added to the set V of the subgraph G that contains the highest number of social ties to that user. Any remaining ungrouped users are discarded. Finally, at Stage 5, the method creates an event associated with each subgraph. Notice the difference between the events from this method compared with those produced by method 1.

The Event Management Process
To incorporate the data fusion methods into an operable system, we designed an Event Management Process. Its purpose is to execute the Data Fusion Process when triggered to do so, and to update the system state following the output from the Data Fusion Process. To ensure that events are managed in real-time from creation to closure, the Event Management Process has three key functions: to create events; to update the state of events; and to end events.

Creating Events: An event is created when the Data Fusion Process outputs an event that does not already exist, i.e. is not linked to an existing event's calendar entry. Events are assigned a start time on creation.

Updating Events: An event is updated when the Data Fusion Process outputs an event that already exists (as indicated by a calendar entry). Both location and attenders are updated to reflect the state of the event.

Ending Events: An event is ended and assigned an end time when it has not been updated for a time period t, the time window in which a user's location update is considered 'live'.

TESTING OUR HYPOTHESIS
To test our hypothesis, we used the location, social network and calendar data gathered during our field study applied to the model in Figure 2.

Co-presence received its sensor inputs from the timestamped Bluetooth location updates and a database of the participants' Bluetooth devices.

Social Network used the participants' Microsoft Outlook contact address books as a virtual sensor input. A social tie between two participants was determined by the existence of one participant in the other's address book, i.e. an undirected edge.

Planning used the timestamped set of Microsoft Outlook calendar events gathered during the field study. This is the original set of 474 events C used in the earlier analysis of the workplace calendar.

We used the timestamped Bluetooth data to simulate real-time operation, so the Data Fusion Process was executed *on every Bluetooth update*, i.e. when any participant's device or static device reported their Bluetooth scan to the server. As mentioned, we did not place a static device in the team desk area (see Figure 1) so participants did not report sightings of static devices, or vice versa, when in this area. Although this

was done to avoid ambiguous location updates, it does mean that latency is introduced into the system. For example, a participant leaving a meeting area to return to her desk will not update her location explicitly. Rather t, the time window in which a participant's location is considered 'live', is used so the system will register her as 'not in any meeting area' after time t has expired. This consequence is important when considering event time results. None of the recorded real-world events were held in this area during the study, but it important to note that, if they had, they would not have been identified.

Results

First, we compare the events output by both data fusion methods against the original calendar events in order to measure any improvement and test our hypothesis. Secondly, we compare the metrics of successfully identified events to their recorded real-world counterparts in order to evaluate the performance of our data fusion methods on this particular dataset. Before comparing the data fusion methods to the calendar alone, we define our notions of successful, false and failed event identifications using the methods.

A successful event identification is defined as a system-identified event that shares an ID with a recorded real-world event. This ID was the unique ID of the calendar event used by the Data Fusion Process to create an identified event (C_1 and C_2 in Figures 3 and 4). We assigned the recorded real-world events the same ID as their corresponding calendar entries in set $G(38)$, so successful event identification was measured by comparing the ID of the system-identified event with the ID fields of the recorded real-world events. In addition, the identified event must be located in the same place at the same time as the recorded real-world event. 'At the same time' means that the time window of the identified event overlapped the time window of the recorded real-world event.

A false event identification is defined as a system-identified event whose ID either i) does not match the unique ID of any recorded real-world event, e.g. the calendar event used to create the identified event in the Data Fusion Process is not a member of $G(38)$; or ii) does match the unique ID of a recorded real-world event but is not located in the same place at the same time.

A failed event identification is defined as a recorded real-world event that has not been identified as an event by the data fusion method. The total number of false event

Metric	Method 1	Method 2
Successful event identifications	37	32
False event identifications	32	14
Failed event identifications	1	6

Table 4. Event identification results

Metric	Method 1	Method 2
Start time (μ_s, σ_s) (nearest 5 min)	(-5, 20)	(0, 15)
End time (μ_e, σ_e) (nearest 5 min)	(-5, 20)	(-5, 20)
Location	0.97	0.84
$\sum c$	112	94
$\sum f$	36	31
$\sum a$	10	29
$\overline{J(V_r, V_i)}$	0.65	0.60

Table 5. Data fusion events vs real-world events: μ = mean, σ = standard deviation. Negative μ indicates the identified time is earlier than the equivalent recorded time

identifications can be calculated by subtracting the number of successful event identifications from the total number of recorded real-world events.

From the results in Table 4 we see that method 1 outputs a greater number of successful event identifications and, subsequently, fewer failed event identifications than method 2 but with a larger number of false identifications. Comparing these results to those of the calendar, we see that our methods reduce the number of false event identifications from 204 to 32 for method 1, and 14 for method 2. This is achieved at a cost of 1 failure of event identification for method 1 and 6 failures for method 2.

Table 5 compares timings of events identified by the two methods and the 38 recorded real-world events, which is comparable with the calendar analysis in Table 2. Once again, c represents a successful attender identification, f a false attender identification and a a failed identification. We have factored in the cost of failed event identifications to each method so, for example, the total sum of failed attender identifications Σf includes the attenders from the recorded real-world events who were not identified.

From these results, we see a minor improvement in identification of the start and end times compared with the calendar alone, as evidenced by the data fusion mean times being equal to zero or closer to zero than those of the calendar. The start time standard deviation σ_s for both data fusion methods improves upon the equivalent for the calendar. However, end time standard deviation σ_e for both methods does not improve upon the calendar. Reasons for this are discussed in the next section. Due to our fusion with location data, we see a significant improvement in location identification for both methods but the mean Jaccard indices are not improved (V_i is the set of attenders identified by the data fusion, and V_r the set of real-world event attenders), also discussed in the next section. Overall, however, we have

Effect	Causes
False event identification	False sensor readings OR Participant mobility
False attender identification	False sensor readings OR Participant mobility
Failed event identification	Sensor failure OR False sensor readings
Failed attender identification	Sensor failure
Event time deviation	False sensor readings OR Participant mobility OR Sensor failure

Table 6. Observed effects and their general causes

shown a significant improvement to the original calendar event set through data fusion, particularly in the large reduction in false event identifications.

DISCUSSION

Here the implications of false and failed identifications are discussed, followed by an analysis of their root causes.

The Significance of False Identifications: Depending on the type of application, false identifications will vary in their significance. If privacy were a critical factor, then they would be very significant: you would not want users added to events that allowed them access to sensitive content intended only for participants in the event. In this case minimising false attender identifications is imperative. Moreover, false event identifications can be seen as a form of spam. Imagine a scenario where two users are walking past each other with a calendared placeholder. A false event may be created around this placeholder since the users are co-present, in each other's contact network and sharing a calendar 'event'. To the users, who in reality are not part of any such event, this could be irritating if, for example, the system attempted to remind them of the event.

The Significance of Failed Identifications: Failure to identify attenders or entire events results in additional burden to users of such a system. If a user were not added to an event they were really part of, then they would have to be manually added. This could become tedious if failures are common. Failure to identify events can lead to further burden: users would have to create the event itself manually, which is onerous.

What Causes False and Failed Identifications?

Here we present a cause and effect analysis of the false and failed identifications in our study. Table 6 lists the effects along with Boolean expressions to represent their causes.

False sensor readings are sensors not reporting the true state of the world. Examples from our study include: i) Bluetooth radio reflections causing devices to see each other when outside normal ranges of coverage, e.g. participants at their desks reported as 'in a meeting area'. ii) Calendar placeholder events and shared reminders. We use the original calendar as a virtual sensor in the Data Fusion Process, and the large number of false events we saw in our analysis manifested as false sensor readings. Thus we saw examples where users were sighted as co-present at the same time as placeholders or shared reminders in their calendar. This greatly increased the likelihood of a false event or attender identification.

We have mentioned the ramifications of choosing not to place a static device in the team desk area, as well as the impact of the 'live' location time window t. We discussed the system's reliance on the expiration of period t to register the participant as 'not in any meeting area'. If we assume t_1 to be the time at which the participant's location is updated while still in the event, and t_2 to be the time the participant leaves the event area, then a false location reading is reported for time $t - (t_2 - t_1)$.

Sensor failure occurred when the sensors did not report data to the system when they should have. We observed the occasional Bluetooth sighting failure, i.e. participants not being sighted when in the ranges of coverage depicted in Figure 1. Occasional connectivity issues were observed when Bluetooth scan results were not reported in real-time. Results that were not communicated were stored locally on the participant devices until a connection was re-established. However, in some cases, the results were reported *after* the event had occurred. It is possible to use this data to create the event *post-hoc*, but real-time functionality is damaged.

We mentioned fuzzy time when considering calendar event selection in the execution of method 2. We requested calendar entries at one particular time, so entries listed *near* that time were not considered. We saw how variable the comparison was between the calendar and recorded real-world event times, so it is possible for a planned event to occur outside its calendared timeslot. Therefore it could be argued that introducing fuzzy time and requesting entries in a time window could capture the calendar entries associated with such events, and help reduce the number of failed event identifications in method 2.

Participant mobility concerns the movement of participants around the study space. Even though we carefully chose the location of the static devices, we observed cases of participants moving into these areas when not involved in events there. An example of this was a participant who would frequently stand in a meeting area making calls on his mobile phone, which was being identified by the Bluetooth scans. Sometimes a relevant event was occurring in the meeting area, an attender of which had a social tie to this participant. The system therefore identified the participant as attending the event, resulting in a false attender identification. This problem also occurred when participants walked by meeting areas with ongoing events; the system would add them to the events if they had social ties to participant attenders.

CONCLUSION

We have shown that a shared office calendar is not a good standalone virtual sensor because it does not represent reality well. As well as genuine events, it also contains reminders and placeholders. Although we saw a good match between real-world event attenders and calendar event attenders, the calendar did not match the real-world event locations or start and end times well.

We have proposed two real-time data fusion methods that combine the calendar with location and social network data to improve the representation of reality given by the calendar alone. Applying the methods to the data gathered during our field study, we improved the number of false events from 204 using the calendar alone, to fewer than 32 using the data fusion methods.

We also enhanced the representation of genuine events, with improvements made to time and location data. We believe that the calendar is a valuable source of context, but the useful data is hidden among false events. Fusion of the calendar with other context sources has shown that the useful data can be uncovered, thereby enabling the development of new applications or improvements to existing applications such as presence and availability systems. In future work, we intend to further analyse our data by breaking the collective dataset down to a 'per-person' basis and to investigate the usefulness of the calendar in alternative environments.

REFERENCES

1. Fogarty, J. Presence versus availability: the design and evaluation of a context-aware communication client. International Journal of Human-Computer Studies 61, 3 (2004), 299-317.

2. Geyer, W., Richter, H. and Abowd, G.D. Towards a smarter meeting record—capture and access of meetings revisited. *Multimedia Tools and Applications 27*, 3 (2005), 393-410.

3. Gütl, C. and García-Barrios, V.M. Semantic meeting information application: a contribution for enhanced knowledge transfer and learning in companies. In *Proc. ICL 2005*, (2005).

4. Horvitz, E., Koch, P., Kadie, C.M. and Jacobs, A. Coordinate: probabilistic forecasting of presence and availability. In *Proc. UAI 2002*, AAAI Press (2002), 224-233.

5. Indulska, J. and Sutton, P. Location management in pervasive systems. In *Proc. ACSW Frontiers 2003*, Australian Computer Society (2003), 143-151.

6. Jones, S. and O'Neill, E. Context-aware messaging: how personal, spatial and temporal constraints affect text-based communication. In *Proc. MUM 2009*, ACM Press (2009).

7. Mynatt, E. and Tullio, J. Inferring calendar event attendance. In *Proc. IUI 2001*, ACM Press (2001), 121-128.

8. Neustaedter, C., Brush, A. and Greenberg, S. "The calendar is crucial": coordination and awareness through the family calendar. *ACM Transactions on Computer-Human Interaction 16*, 1 (2009), 1-45.

9. O'Neill, E., Kostakos, V., Kindberg, T., Schiek, A.F., Penn, A., Stanton Fraser, D. and Jones, T. Instrumenting the city: developing methods for observing and understanding the digital cityscape. In *Proc. UbiComp 2006*, ACM Press (2006), 315-332.

10. Palen, L. and Grudin, J. Discretionary adoption of group support software: lessons from calendar applications. *Organizational implementation of collaboration technology*. Springer-Verlag, (2002).

11. Payne, S. Understanding calendar use. *Human-Computer Interaction 8*, 2 (1993), 83-100.

12. Pentland, A. and Eagle, N. Social serendipity: mobilizing social software. *Pervasive Computing 4*, 2 (2005), 28-34.

13. Schmidt, A., Asante Aidoo, K., Takaluoma, A., Tuomela, U., Van Laerhoven, K. And Van de Velde, W. Advanced interaction in context. In *Proc. HUC 1999* ACM Press (1999), 89-101.

14. Tullio, J. and Mynatt, E. Use and implications of a shared, forecasting calendar. *Lecture Notes in Computer Science 4662*, Springer-Verlag (2007), 269-282.

15. Tullio, J., Goecks, J., Mynatt, E.D. and Nguyen, D.H. Augmenting shared personal calendars. In *Proc. UIST 2002*, ACM Press (2002), 11-20.

16. Tullio, J. Exploring the design and use of forecasting groupware applications with an augmented shared calendar. Ph.D Thesis (2005).

17. Wang, J., Chen, G. and Kotz, D. A sensor-fusion approach for meeting detection. In *Proc. MobiSys 2004*, ACM Press (2004), 1-7.

18. Westermann, U. and Jain, R. Toward a common event model for multimedia applications. *Multimedia 14*, 1 (2007), 19-29.

19. Xie, L., Sundaram, H. and Campbell, M. Event mining in multimedia streams. In *Proc. of the IEEE 96*, 4 (2008), 623-647.