The Carrot Approach: Encouraging use of location systems.

Kieran Mansley, Alastair R. Beresford and David Scott.

Laboratory for Communication Engineering,
Cambridge University Engineering Department,
William Gates Building, 15 J.J.Thomson Avenue,
Cambridge CB3 0FD. UK
{kjm25,arb33,djs55}@cam.ac.uk

Abstract. The Active Bat system provides the ability to locate users and equipment with a high degree of accuracy and coverage. Despite this, participation is low. We are concerned that this is symptomatic of a fundamental problem in location-aware computing; specifically the lack of understanding about which applications are useful and what factors motivate people to use them.

In this paper we provide a retrospective analysis of Bat system usage grounded in game theory. We have analysed the needs of people within the coverage area, and used this to motivate a set of highly targeted location-aware applications which we believe are compelling enough for individuals to induce a gradual increase in participation. This *carrot* approach has been successful and has increased the number of people who wear their Bat.

Finally, this paper provides a critique of our experience with the Active Bat system. We suggest a number of refinements that should be considered by developers of future location systems.

1 Introduction

1.1 The Active Bat System

The Active Bat [1] system was originally deployed at AT&T Laboratories Cambridge, and later at the Laboratory for Communication Engineering (LCE). The Bat itself is a small battery-powered device (Figure 1 *left*) with a radio transceiver, ultrasound transmitter, two push-button switches (which signal their depressed state over the radio link) and an audible speaker pre-programmed to play one of twelve different sounds (again, when triggered to do so by radio communication). The installed infrastructure includes a radio transmitter which schedules Bats to transmit a pulse of narrow-band ultrasound which is in turn received by a matrix of time-synchronised ceiling receivers; the time of flight data to each ceiling receiver is used to multi-laterate the location of the Bat to within a few centimetres.

In order to write applications which use Bat location data, receive Bat button press events and trigger audio sounds, an extensive software architecture

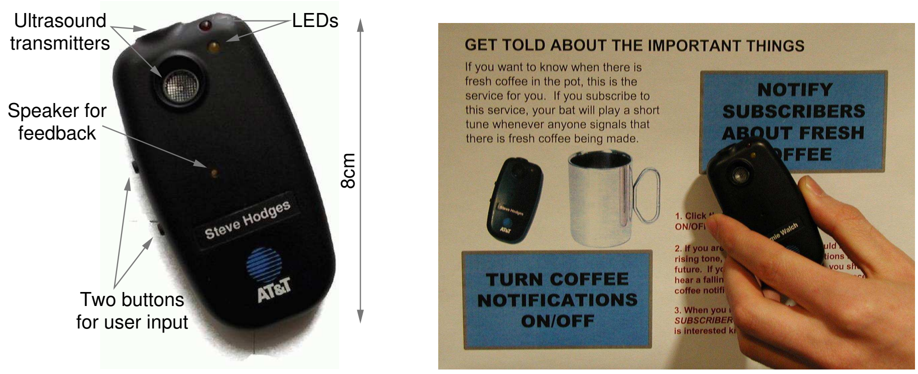

Fig. 1. The features of a Bat (*left*), and a Spatial Poster (*right*)

has been constructed. SPIRIT [2] maintains a world model of the location and state of various Bat-tagged and passive objects (such as people, laptops, computers, desks and chairs) and facilitates *programming with space* via *containers*, which are spatial regions defined by closed polygons and attached to either the floor plan or Bats. Applications can be written in an event-driven style, and receive callbacks whenever containers intersect, subsume or separate from other containers.

One of the innovations at AT&T was the development of *spatial buttons* (Figure 1 *right*) as a means of interacting with the applications. A spatial button is a region of space which has some action associated with it, similar in concept to Want et al's RFID-based augmenting tags [3] but they are completely passive pieces of paper: it is the space they occupy that is important. A user *clicks* on a spatial button by placing their Bat in that space and pressing one of the (physical) buttons located on the side of the Bat. A small poster placed at the same physical location as the spatial button allows the user to discover what the button does (and where it is!). These spatial buttons form an integral part of many of the applications of the Active Bat system: they are a simple, easy to create and use, and offer an intuitive user interface.

1.2 Usage and Applications

Increasing usage of a location system is desirable for a number of reasons: (1) the more people who participate, the more useful some applications become;[1] (2) increased usage affords increased anonymity (this is discussed in Section 3.3); and (3) gathering more data enables more data analysis and research. When the Active Bat system was installed at AT&T, the percentage of employees wearing a Bat was high (around 80%); however its installation at the LCE has resulted in a

[1] This is analogous to Metcalfe's Law: *The total value of a network where each node can reach every other node grows with the square of the number of nodes.*

far lower uptake of users (around 20%) in spite of the fact that both installations included very similar applications.

There are three discernible differences between the installations which may explain the lower level of participation in the LCE: (1) the coverage area of the Bat system at AT&T was larger (covering an entire building) leading to a more useful "person-locator" application; (2) participants at AT&T were employees while the LCE is a university lab; and (3) at AT&T, Bats would automatically unlock internal doors between different parts of the building. These differences shall now be considered individually.

First, consider the case of increased location system coverage leading to a more useful "person-locator" application. From a game-theoretic standpoint [4], this application may be modelled by a multi-player prisoners' dilemma. In real-life each person chooses whether to wear their bat or not whereas in the prisoners' dilemma each prisoner chooses whether to co-operate with the authorities or not. Both wearing a Bat and co-operating have an associated (small) cost. If everyone co-operates (i.e. everyone wears their Bats) then the whole group receives a benefit. However, from the point of view of an individual it is always better not to co-operate (i.e. not wear their Bat) while secretly hoping that everyone else does; this is said to be the *dominant strategy*. It does not matter how great the benefit (i.e. the size of the coverage area) is; if all the players are rational then *no-one* co-operates and *no-one* wears their Bats. Therefore coverage area and applications like the "person-locator" cannot explain the difference in uptake between the LCE and AT&T.

The second difference stems from the fact that AT&T was a company whereas the LCE is a university research lab. AT&T was a stable community in which people were encouraged to act selflessly. In contrast the LCE is a University research lab with a high turnover of personnel and not as closely-knit; in fact the LCE has several small social subgroups. These cultural differences could contribute to the difference in uptake between the LCE and AT&T.

The third difference between AT&T and the LCE—the presence of the automatic door opener in AT&T—is noteworthy in that it provides a useful function to an individual even if *no-one* else wears their Bat. In game-theoretic terms, users who wear their Bats always receive a useful payoff irrespective of the actions of others. For this reason we believe the automatic door opener was the first (albeit accidental) "killer app" of the Bat system. It is important to note that at no point in either deployment have any users been mandated to wear Bats: only *carrots* (i.e. positive inducements); no *sticks* (i.e. punishments) have been used.

In order to promote increased Bat usage within the LCE, and provide a mechanism to bootstrap the system from the low-usage state to the high-usage state, we set out to develop a set of location-aware applications with a view to finding similar "killer apps" for the new Bat environment. Unlike in AT&T, in the LCE we assume that everyone acts in their own self-interest or in the interest of a small group and therefore we must make it in the interests of these individuals and groups to participate. We model the utility to an individual of an application by the formula $Utility = AU^2 + B$ where U is the number of participating users and A and B are constants. AU^2 is the *Metcalfe-effect* and B the single-user payoff. Applications fall into one of three categories: TYPE I: those useful to isolated individuals (high B); TYPE II: those useful to small subgroups (high A, small set of users U); and TYPE III: those only useful when the whole lab participates (high A, whole lab U). Many traditional applications (e.g. the "person-locator") are TYPE III applications; most of the applications we present here are either TYPE I or TYPE II.

1.3 Relevance to Future Location Systems

The Active Bat system was deliberately designed to be more accurate than is likely to be practical in a commercial location system. As the Active Bat system offers excellent location information across a large indoor area, it constitutes a platform for investigating questions such as "when and where is such accuracy or coverage necessary?" Such analysis should enable a more cost effective location system to be constructed that still supports applications prototyped with the Active Bat system.

A lot of money is currently being spent building large-scale location systems, based on technologies such as GPS positioning, mobile phone triangulation, and RFID. We are concerned that the lack of use of the Bat system is not because of lack of accuracy or coverage (as both are excellent) but is symptomatic of a more fundamental problem in location-aware computing; specifically the lack of genuinely useful applications and a strategy for their deployment. We believe that unless this critical stumbling block is removed it is highly likely that no-one will use the less accurate, expensive, wide-area systems either.

Much of the ubiquitous computing community is currently developing and deploying architectures to record and distribute location information in an efficient and cost effective manner. We hope to provide some insight into what works (and does not work) for both application programmers and end users, and how some simple changes to the technology could make some applications much better. It has been asserted that there is a dearth of useful location-aware applications because with current middleware systems they are too hard to develop [5]; we contend that this is not true in our case (after the initial fixed cost of installation) and that developing applications which concentrate on benefits for the wearer are the key to making ubiquitous computing successful.

The rest of this paper is structured as follows. Section 2 presents a description of some of the applications themselves, while Section 3 discusses the suitability of the programming model and the benefits and limitations of the current implementation of the Bat system from the perspective of both the application writer and user. Section 4 presents some measurements and results of the effect caused by the deployment of these applications. Related work (and in particular previous Bat Applications) are discussed in Section 5; finally Section 6 concludes.

2 The Applications

In this section we begin by developing a classification of the intended users of the location system (in our case, the staff and students within the lab) with the aim of targeting applications at the needs of specific social groups.

2.1 Classifying the Users

The first task in any business is to identify potential customers who have needs which can be addressed by a feasible product. A common mistake (especially for technical people) is to focus mostly on the product itself rather than the needs of the customer—a condition known in business circles as *marketing myopia*. Therefore after a period of consultation, observation and informal chat we divided the population of the lab into the following groups (a process often known as *market segmentation*):

Drinkers : lab members who share the industrial-sized lab coffee machine
Players : a group who play online multi-player games (e.g. Quake III)
Teachers : faculty and students who have teaching responsibilities
Collaborators : those involved in joint projects with other researchers
Socialisers : those who regularly attend lab meetings and talks[2]
Everyone : the whole lab

Note that all lab members fall into at least one group and most people fall into several different groups. Each group was targeted with at least one (Type I or Type II) application tailored specifically for the needs of that group. It is important to note that each application was designed to be stand-alone and useful for the group concerned even if no other groups wear their Bat; this is the key to how we avoid the bootstrapping problem described earlier. In addition to each application providing users with more of an incentive to wear their Bats, a large number of total applications represents a good way to hedge our bets; in other words the failure of a small number of applications does not endanger the whole project.

The following table lists the applications for each group:

Group	Applications
Drinkers	Fresh Coffee Notification Service
Players	Capture the Flag, Batdar, Bat-Miles
Teachers	Teaching Timer
Collaborators	Visitor Interface, Office-Watcher
Socialisers	Meeting Reminder
Everyone	Daily Diary

[2] Even though technically many of these meetings are compulsory many people in practice fail to attend for various reasons.

These applications will now be described in the following sections, grouped together by implementation technique: (1) pager applications where the Bat is used to provide audible reminders of events (which are either explicitly signalled by other users, or inferred from the location data); (2) tracking applications which passively process the location data; and (3) games or other applications where users not only receive information about the virtual world, but also generate updates to virtual objects in the world model.

2.2 Intelligent Paging

The input and output facilities of the Bat are heavily restricted in order to attain long battery life (typically 18-24 months). Pager applications are suited to this restricted environment and allow users to be notified by a short audible sound that an event of interest has occurred. In order to prevent uninterested parties from receiving irrelevant audible alerts, users subscribe (or unsubscribe) to a notification service via spatial buttons (as illustrated in Figure 1 *right*) placed in the environment. A rising or falling trio of tones provides feedback on the status of user subscription whenever a user clicks on a spatial button. The printed paper poster located at the same physical location as the spatial button provides additional visual information about the service to the user.

Many of the pagers discussed below do not rely on high-accuracy location information and therefore could be realised using much simpler technology. Nevertheless we claim that, simple as they are, they are still genuinely useful and will motivate people to wear their Bats. As more people start to wear their Bats, other (Type III) applications (like the "person-locator") become increasingly useful. Furthermore, we can use the highly-accurate location information of the user to improve the paging services in a number of ways that would not be possible with simpler technology:

– Notifying only one subscriber per room (reducing the audible distraction resulting from a set of Bats beeping simultaneously).
– Inferring the existence of gatherings or meetings (see Section 2.3) and using this information to delay (or modify in some other way) the notification until the meeting has ended.
– Triggering events automatically from passive location monitoring.

This last point leads us to divide this category of applications into those which require users to actively notify subscribers that an event has taken place by clicking on a spatial button, and those which are able to automatically infer an event.

Active Notification A simple example is the fresh coffee notification service, a Type II application. Fresh coffee is generally regarded as superior to coffee that was brewed some time ago and the discerning drinker likes to know when coffee has been made or at least how long the coffee has been brewing.[3] Users can subscribe to the service by clicking on a spatial button. When brewing is complete a second spatial button is used to notify coffee fanatics (the **Drinkers**) that fresh coffee is available. Data are retained (and viewable on the intranet) so the age of the latest brew is always available. Similar services exist to support notification of food and nibbles placed in the kitchen, and a notification of the start of group meetings (which has had a noticeable impact on the number of late and missing students).

Another illustrative example is our visitor interface, a TYPE I application (intended to be of use to the **Collaborators**): at the main entrance to our laboratory we have a flat panel touchscreen allowing visitors to specify who they have come to see, and discover where that person is (or if they are out). The service notifies the user that a visitor wishes to meet with them, allowing users to know when visitors arrive, regardless of where in the laboratory they happen to be. It also displays a photo of the visitor on the nearest (to the user) Broadband Phone.[4] Subsequent location information is then displayed to the visitor via the flat panel display, providing feedback on the progress of the notified user toward them.

Context-Aware Paging We are able to use the location information available to infer when events have occurred, without requiring the users to signal them explicitly. To illustrate this, consider the modern equivalent of a knot in a handkerchief for the technologically minded (wo)man-about-town. People frequently need to remember to do a task not at a fixed time, but at the same time as an abstract action. For example: *there is something I want to take out of the fridge when I leave work*, or *there is something I have to do when I go for lunch*. A diary is not always appropriate as the reminder is not based on a fixed time, rather on the context of the user (e.g. *when I am leaving the office*); furthermore a diary needs to be read to be effective, whereas our solution is more proactive.

The simplest application is "remind me when I leave the building": by monitoring the user's movement vector as they approach an exit we can determine when they are leaving the building and provide an audible alert.[5] This TYPE I application has been extended to not only provide audible notification of events, but also execute a configurable task. This extension effectively makes it a context-aware version of the Unix `cron` command scheduler.

A number of graduate students teach small groups of undergraduates for hour-long sessions (the **Teachers**). To prevent sessions over-running and ensure that teaching is not disturbed (or the room double-booked), a spatial button can be used to notify the meeting room management system that a teaching

[3] The Computer Lab in Cambridge has a long history of using innovative technology to disseminate this information [6].

[4] *Broadband Phones* are telephones that incorporate a large display that operates as a stateless thin-client running VNC [7], allowing a wide variety of applications to be run on a server and displayed on the phone.

[5] A number of students use this just to remind them to take their Bat off.

session is about to take place. This Type I application marks the meeting room as busy on the lab intranet and provides an audible alert to the teacher's Bat a few moments before the teaching time is up.

The office-watcher (Type II) application provides a spatial button for every member of our laboratory next to their own office door. Users who attempt to visit a colleague (the **Collaborators**) and find they are not in can click the office-watch spatial button and receive an audible alert when the person they are interested in next returns to their office. Users can also click their *own* office-watcher button to receive an audible reminder to do something when *they* next return to their office.

2.3 Tracking Applications

This category differs from the Pager Applications in Section 2.2 in that the applications do not notify the user of anything. Instead, they passively gather information for later presentation. Since the user interface of the Bat system is minimal, these applications use the lab intranet to provide a better (visual) display of pertinent data.

Bat-Miles[6] is a Type I application. Users subscribe (or unsubscribe) using a spatial button; subscribed users receive details of their distance travelled while standing and sitting, maximum speed and approximate calories consumed.

A personalised daily diary (a Type I application) of user events is constructed from the services users subscribe to and includes times at which coffee has been made, supervisions given and photographs of visitors to the front door (taken with a web camera). Details of the times and locations of meetings with other lab members can be automatically inferred [8] and recorded. Figure 2 shows an example screen shot of a daily diary of events.

2.4 Games

Historically games have been a major driving force behind developments in computer technology, and judging by the number of people in the lab who play games regularly (the **Players**) this trend looks likely to continue. We believe that location-aware applications which are fun will both boost participation while pushing the technology to its limits. The games are multi-player and are therefore Type II applications.

Counter Strike and Capture The Flag As real-life is, almost by definition, experienced from the first-person, a first-person "shoot-'em-up" is the obvious candidate: traditional games of this type try to *emulate* the first-person perspective whereas the Bats provide *direct access* to this view point, and so provide a natural interface. The game had two design requirements: (1) the experience of playing game by physically moving around the building should be a feature

[6] Bat-Miles is named after Air-Miles loyalty schemes.

Fig. 2. The Daily Diary Log.

rather than a hindrance; and (2) usage of additional hardware should be minimised so that players only need to carry their Bats. With these requirements in mind we settled on two games which are relatively simple (and similar) in concept:

Counter Strike: The players are divided into two teams: *Terrorists* and *Counter-Terrorists*. The terrorist's aim is to plant a bomb at a pre-defined location, and the counter-terrorists must try to stop them.

Capture the Flag: Two teams each try to retrieve a "flag" from the other team's base, returning with it to their own base. At the same time they must prevent the other team from capturing their own flag.

Traditionally, games of this sort (e.g. Quake III) involve a large amount of weaponry. It was clear that the Active Bat system would struggle to support these sorts of fast moving interactions where choice of weapon, aim (and so orientation) and latency of position information is critical. We therefore opted to support two weapons: land-mines and shotguns.

Players can join either team by going to the relevant base and using a spatial "join" button. This button is also used to rejoin the game if the player dies. The natural overhead of physically returning to your team's base to be reincarnated is sufficient to make death a hindrance.

All of the objects in the game (flags, bombs, shotguns, and land-mines) are entirely virtual. The two weapons are both deployed using the same button on the Bat and the weapon the player intends to use is determined using *gestures*. Land-mines can be placed by bending down and clicking near the floor with the Bat, while the shotgun is fired by clicking twice (to indicate the direction of the shot) at chest level.

Flags and bombs are picked up and put down automatically if the player is in the correct location (i.e. one of the bases). Land-mines are triggered when a user steps within a certain proximity of one once they have been armed, and so cannot be picked up. Each player starts the game with a fixed number of mines, but these are automatically replenished once they have exploded.

All the feedback to the users is done through audible signals. This includes both sounds from the Bat (to acknowledge events that refer to an individual player, such as "you have died" or "you have laid a mine") and from any PCs around the building that are running a special client that listens for game events. Game events are generally things that affect a whole team, such as "your team has won/lost", "your team has the enemy flag", etc. This works well, as the speaker on the Bat is low in volume, localising the sounds about individual players to themselves, while the speaker on the PC is high in volume, broadcasting events of general interest to many players. We found placing one of these PCs in each base led to sufficient feedback to make the game playable. Each PC can also be configured to display a map (Figure 3) of the game area, allowing players in the bases (or just spectators) to see a view of the virtual world. At any point in time a user is able to query their status (alive or dead, and what they are carrying) using the other Bat button, and the result is again conveyed using audible beeps.

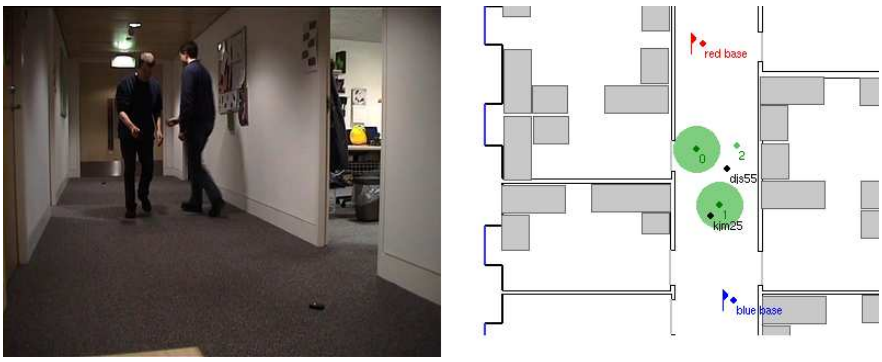

Fig. 3. The Capture the Flag Game
Two players ("djs55" and "kjm25") are playing the game. The physical world is shown on the left, and a plan view of the corresponding virtual world on the right. The player "kjm25" has just triggered the mine labelled "1" (which in turn set off mine "0"), and has been caught in the blast zone.

These two location-based games demonstrate that it is possible to use a device with very limited feedback capabilities to support games that are genuinely fun to play, and combine elements of the physical world (e.g. moving from one place to another) with those of the virtual world (e.g. weapons) [9]. We believe that its capabilities could easily be extended by providing a richer environment (e.g. augmented reality), or more weapons that are controlled with different gestures.

Batdar While a Bat enables its wearer to be located, Batdar[7] uses the Bat to help the wearer locate other objects. It makes the Bat beep with a period proportional to how close it is to something, in the same way that modern car reversing aids do. It can use a number of metrics to determine the frequency of the beeps: (1) how far away the target is; (2) how much the Bat's orientation differs from the bearing to the target; or (3) how much the Bat's direction of travel (or the direction of a particular gesture) differs from the bearing to the target.

Of these choices the latter seems to be the most effective; the "distance to target" fails when the user is a long way from the target: they have no indication which way to start looking, and the orientation metric fails as orientation accuracy from a single Bat is insufficient. Whilst Batdar can be used to support guiding visitors to a destination, the limitations in user feedback (described in more detail in Section 3) can make this exercise frustrating; a simple map is much more effective at conveying the same information. Batdar is more suitable for "treasure hunt" games where users must seek out a particular target or targets within a certain period of time.

3 Performance of the Active Bat system

The SPIRIT system uses CORBA to connect location-aware applications with live location data and persistent data which is stored in an Oracle database. At the LCE a set of Python language bindings have been implemented to allow rapid application development. The ease with which we developed and deployed the applications described here has encouraged other users of the system to develop their own applications (e.g. using the Bat to notify a user when new email has arrived). In the research environment, having a complete and well-written middleware system for users to take advantage of can be seen as another *carrot*.

Location information available from the Bat system is generally of excellent quality and most aspects of the Active Bat and SPIRIT system have been well thought-out. However a number of compromises had to be made in order to guarantee a long Bat battery life. In this section we discuss our experiences with developing context-aware applications with the Bat system and which areas should be rethought in a future location system design.

3.1 Feedback Issues

The weakness that came to light almost immediately was the limited number of ways that we could provide feedback to the user. The Bats have no means of

[7] Batdar is named to allude to SONAR.

providing visual cues (other than two LEDs that indicate use of the radio channel) and so feedback relies on the use of a speaker to provide audible tones. This minimal interface stems from the requirement for long battery life (which in turn was chosen to minimise the cost to the user of wearing the Bat). Future location systems (such as mobile phone networks) whose devices require regular charging anyway (e.g. mobile phones) should be able to have much more sophisticated display facilities.

Each Bat is pre-programmed with twelve tunes which range from simple beeps to more complex sequences similar to mobile phone ring-tones. To achieve a degree of consistency many of these tunes are used universally to signify the same thing. For example, a series of rising tones is used as positive acknowledgement (e.g. "you have turned something on") and a series of falling tones is used as negative acknowledgement (e.g. "you have turned something off").

The remaining tunes are then used to signify a wide range of events. We have experimented with using single pre-programmed tunes that have an obvious association ("Food, Glorious Food"), and combining some simple beeps together to form more significant sequences. In theory messages could be encoded using Morse code or a similar message structure to provide additional meaning.[8]

Combining the existing beeps together to make sequences reveals a second limitation of the Bat system. Triggering a Bat beep requires the use of the radio link to the Bats, consuming considerable battery power; therefore the radio channel is normally turned off whenever it is not required. The interval at which the Bat wakes up (to perform location updates and play audible tones) is highly variable, and so it is difficult to trigger beeps at a high rate. This is illustrated in Figure 4 where (for a stationary or slow moving user) you cannot reliably trigger more than 1 beep per second. If the user moves at a reasonable speed, the achievable update rate increases. Pressing a Bat button also forces an update over the radio channel (and thus removes any latency in user-directed actions). The Bat system provides a partial solution, in that it is possible explicitly request that the Bat be prevented from going to sleep: it forces the Bat into a *high-rate* QoS mode. However, this reduces battery life, and thus should only be used for short periods. Finally, putting too many users into *high-rate* mode overloads the radio channel and reduces the responsiveness of the audio feedback for other users.

With many applications using the speaker on the Bat to convey information, the user can have difficulty attaching meaning to sounds. This difficultly motivated the construction of the user's daily diary application described in Section 2.3 which allows the user to quickly answer the question "why did my Bat just beep?". Web pages were also used for feedback in the tracking applications where the amount of information was too great to feasibly present without a screen. Even a small screen (such as a mobile phone display) may have been able to support these applications (for example, compare the Bat-Miles application with bicycle odometers).

[8] Mobile phone users may be familiar with this, as the delivery of a text message is often signalled using the Morse for SMS.

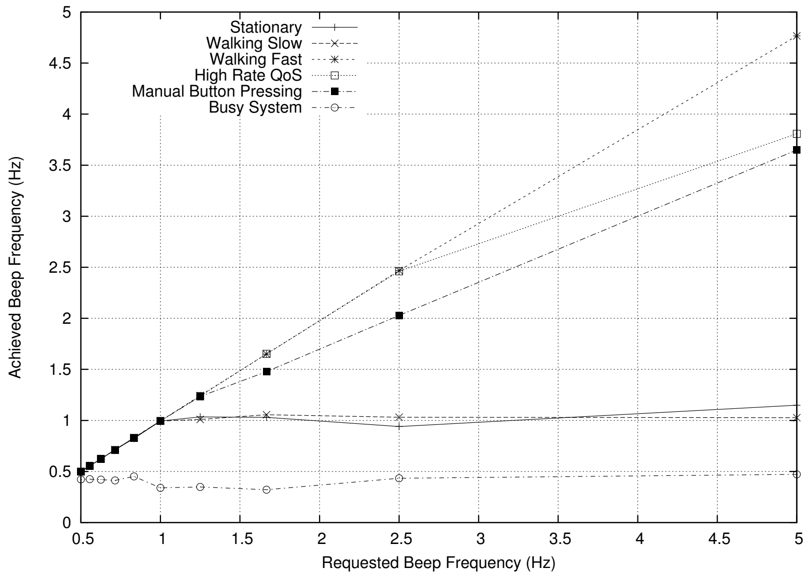

Fig. 4. Audible Feedback Rate of the Bat System.

3.2 Accuracy of Location Data

Many of the applications described in this paper can be implemented using location systems with smaller coverage area or less location accuracy when compared with the Active Bat system. For example spatial buttons could be implemented using visual tag recognition and Bluetooth capabilities of a mobile phone handset [10] or by embedding RFID readers into the spatial button posters or even clothing [11].

The games were the most demanding on coverage area and resolution, and gesture information was used to overcome the inaccuracies in orientation information available from a single Bat. Location systems with greater orientation accuracy would therefore allow a more expressive location-based gaming environment. A high update rate is needed for the games to be playable and fun (if a person is only located every few seconds, interaction with other places and virtual objects, such as mines, becomes intermittent and frustrating). Therefore interactive location-based gaming requires a method of achieving a high quality of service (e.g. the Active Bat system's high-rate mode).

3.3 Privacy

A common difficulty with location-based systems (particularly one as pervasive, centralised and accurate as the Active Bat system) is that they can be very privacy invasive. Some of the applications developed here (e.g. coffee alerts) can be made to work without knowledge of the true user identity, and with some work could be made to function with untraceable pseudonyms (identifiers an attacker cannot associate with any real-world entity). It is possible that anonymising information in this way can prevent an application from determining the true user identity hiding behind the pseudonym [12].

Some applications will always require knowledge of user identity (e.g. the Office-Watcher). Our current solution is to build *reciprocity* into the system—providing feedback via the diary intranet page of requests for location information to the location data owner. This has the effect that frequently watched users can find out who is watching them and take appropriate action if the service is abused. Economists have noticed that adding punishments (as well as rewards) has a positive effect on the outcomes of competitive games [13].

4 Measurements and Results

To quantify the effect of this work we distinguish users who wear their bat from users who do not by performing an off-line analysis of the centralised log of all the location sightings each day. A person is deemed to have worn their Bat if it is sighted in three different rooms and one of the rooms is the central corridor of our laboratory.[9] There are clearly ways in which this could be spoofed (by getting someone to wear more than one bat), and it would not count people who only wear their bat in their office, but, after careful manual observation of user habits, both these outcomes occur very infrequently.

One of our goals for this work was to demonstrate that by providing enough *carrots* in the form of useful applications (and without using any *sticks*) we could encourage users to wear location devices.

By carefully choosing which applications to implement based on an informal lab survey and avoiding the pitfall represented by the Prisoners' Dilemma, this goal has been achieved, as Figure 5 demonstrates. Bat usage is up from an average of 5.2 people (6.8 on weekdays) for the six months prior to the deployment of the new applications to 13.1 (16.3 on weekdays) for the following six weeks. Quantifying the precise level of improvement is difficult since the sample size is small, however, if the difference in occupancy rate of a student lab (like the LCE) and a commercial lab (like AT&T), is taken into account we now have a comparable percentage of users in both deployments.

Despite the increase in usage, a significant minority of people still do not regularly wear their Bat. An informal survey threw up a number of reasons: (1) apathy or forgetfulness—they just do not feel it is worth the effort or forget to put it on; (2) privacy—there are some who are concerned about their privacy, and so do not wear one on principle; and (3) discomfort—some people find wearing a Bat around their neck uncomfortable or annoying.

The group represented by (1) could potentially be encouraged by targeted (Type I) applications that they would find useful.Other ways of wearing a Bat

[9] These requirements avoid counting people who just leave their Bat on their desk, and make counting erroneous sightings (due to other sources of ultrasound) unlikely.

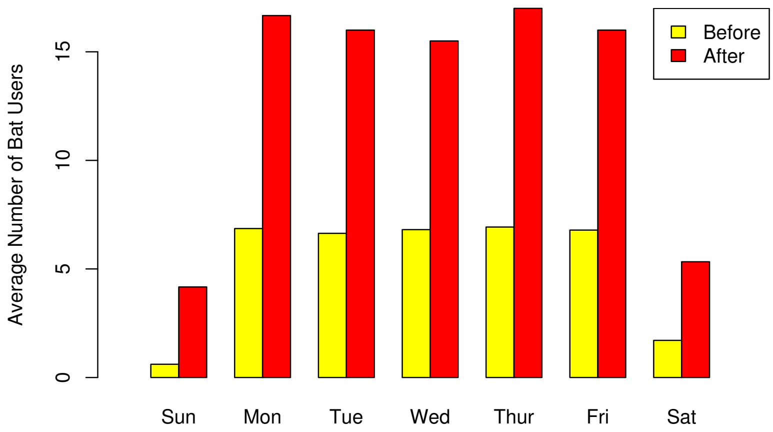

Fig. 5. The Number of People Wearing Bats.

(such as belt clips or watch straps) are made available to try to resolve the problems of group (3). The group who are concerned about their privacy are unlikely to be persuaded to use the Bat system in its current form, and this group demand a heavily re-designed location system in order meet their privacy concerns.

The popularity of the different applications varied considerably. Given the different usage patterns for each application and their concurrent deployment it is difficult to formally compare them and measure which has had most impact. However, those that have the most subscribers are the fresh coffee (and food) notification pagers, and the Bat-Miles scheme. Perhaps unsurprisingly these are ones for which the users get a tangible benefit with little effort.

Finally, it is clear that novelty value has a large impact on the number of people who wear their Bats. The Active Bat system has now been deployed in two labs, and in each there was a gradual tail off in usage (before stabilising) after an initial high take up. This peak was recovered by the deployment of new applications, but without a continual output of new features[10] it is likely to decline again in the future (although stabilising at a higher level). In an attempt to avoid novelty being the major factor in our measurements, the applications were not hyped or pushed to the users: their presence was announced by email, and by a short presentation at a lab meeting.

[10] We have started to observe that the once the system reaches a certain usability, these new features come from applications developed and deployed by the users themselves

5 Related Work

AT&T Labs Cambridge developed a set of location-aware applications as part of their initial deployment of the Active Bat system. Some applications were adapted from the original Active Badge [14] system; for example: locating users, unlocking of internal doors, forwarding phone calls and teleporting desktops using VNC [7]. New applications developed specifically for the Bats included spatial buttons to provide a simple and intuitive interface to control a "sentient" scanner and a method of using the Bat as a mouse pointer on large plasma screens installed at various locations throughout the building.

Presentations given by researchers at AT&T were context-aware—slide shows could be controlled with a Bat, and meeting details were recorded and published on the lab intranet (this system was similar to some of the methods proposed for the Reactive Room [15]). Video conferencing was also context-enabled: the delegates present in a conference room were tracked to select the most useful camera angle automatically.

Other office applications and systems have been proposed, including the stick-e note architecture [16] which allows applications to attach virtual documents to physical spaces. Collaborage [17] used image processing techniques to allow physical notes (e.g. in/out and away boards for employees and "to-do" lists) to be synchronised with their virtual counterparts.

Outside the office environment, GPS systems have been used to build everything from guidance systems for the blind [18] to location-based automated assistance for archaeology fieldworkers [19]. Tourist guide applications are perhaps the most popular with ubiquitous computing researchers; examples include the CyberGuide project at Georgia Tech. [20] and the Lancaster GUIDE project [21].

In the field of ubiquitous computer games a number of projects have been developed [22] to see what existing game functions can be supported with current ubiquitous technology [23], and what new games can be developed to take advantage of possible future advances in ubiquitous technology [24].

6 Conclusions

In this paper we have shown how usage of location systems can be incrementally increased by taking a more business-like, customer-focused approach to application development. We analysed the recent decline in Active Bat usage from a game-theoretic standpoint and argued that many existing location-aware Type III applications have fallen into disuse as a consequence of the well-known Prisoners' Dilemma. We described how this trap could be avoided if Type I or Type II applications are provided which are of immediate use to individuals and small social groups. Furthermore, increased overall participation has an overwhelmingly positive effect as users of the location system receive a community benefit from increased take-up, both from being able to locate colleagues more reliably and from increased privacy. We claim this principle justifies the existence of applications that have no intrinsically useful purpose (such as games).

Due to the limited nature of most location system tags, interaction with the user (rather than location accuracy and coverage) is the current limiting factor in application development. We have shown how best use can be made of a device with limited feedback capabilities, and how alternative channels can be used to provide parallel feedback to the user.

In conclusion we believe that designers of future location systems must urgently consider both the interaction capabilities of their locatable tags and carefully analyse the features required by the applications users actually *want* if they are to have a successful, popular and profitable system.

Acknowledgements

Thanks go to Alastair Tse for building the Python-SPIRIT interface library, Rob Harle for his help with the Active Bat system, Andy Rice for his comments and work on the visitor interface, and Anil Madhavapeddy and Rob Headon for their helpful comments on drafts of this paper. Finally, a big thank you to all those AT&T Labs Cambridge employees who designed and built the Active Bat system and associated SPIRIT middleware.

References

1. Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster. The anatomy of a context-aware application. In *5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom '99)*, 1999.
2. Mike Addlesee, Rupert Curwen, Steve Hodges, Joe Newman, Pete Steggles, Andy Ward, and Andy Hopper. Implementing a sentient computing system. *IEEE Computer*, 34(8):50–56, August 2001.
3. Roy Want, Kenneth P. Fishkin, Anuj Gujar, and Beverly L. Harrison. Bridging physical and virtual worlds with electronic tags. In *SIGCHI conference on Human factors in computing systems*, pages 370–377, 199.
4. Robert Gibbons. *A primer on Game Theory*. Prentice Hall, 1992.
5. Anthony Lamarca and Maya Rodrig. Oasis: An architecture for simplified data management and disconnected operation. In *17th International Conference on Architecture of Computing Systems — Organic and Pervasive Computing (ARCS'04)*, March 2004.
6. Quentin Stafford-Fraser. The life and times of the first web cam. *Communications of the ACM*, 44(7), 2001.
7. Tristan Richardsonm, Quentin Stafford-Fraser, Kenneth R. Wood, and Andy Hopper. Virtual Network Computing. *IEEE Internet Computing*, 2(1):33–38, January 1998.
8. William M. Newman, Margery A. Eldridge, and Michael G. Lamming. Pepys: Generating autobiographies by automatic tracking. In *Second European Conference on Computer-Supported Co-operative Work (CSCW)*, pages 175–188, September 1991.
9. Kieran Mansley, David Scott, Alastair Tse, and Anil Madhavapeddy. Feedback, latency, accuracy: Exploring tradeoffs in location-aware gaming. In *3rd Workshop on Network and System Support for Games (NetGames), part of ACM SIGCOMM 2004*, August 2004.

10. Anil Madhavapeddy, David Scott, Richard Sharp, and Eben Upton. Using cameraphones in augmented reality applications. Submitted to ISMAR 2004, available from `http://www.cl.cam.ac.uk/Research/SRG/netos/uid/spotcode.html`.
11. Robert Headon and George Coulouris. Supporting gestural input for users on the move. In *Proceedings of Eurowearable '03*, pages 107–112, September 2003.
12. Alastair Beresford and Frank Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, March 2003.
13. James Andreoni, William Harbaugh, and Lisa Vesterlund. The carrot or the stick: Rewards, punishments, and cooperation. Available from the University of Oregon Economics Department at `http://econpapers.hhs.se/paper/oreuoecwp/2002-01.htm`.
14. Roy Want, Andy Hopper, Veronica Falcão, and Jonathan Gibbons. The Active Badge Location System. *ACM Transactions on Information Systems*, 10(1):91–102, January 1992.
15. Jeremy R. Cooperstock, Signey S. Fels, William Buxton, and Kenneth C. Smith. Reactive environments. *Communications of the ACM*, 40(9):65–73, September 1997.
16. P. J. Brown. The stick-e document: a framework for creating context-aware applications. In *Proceedings of Electronic Publisher*, number 2 in 8, pages 259–272, January 1996.
17. Thomas P. Moran, Eric Saund, William Van Melle, Anuj U. Gujar, Kenneth P. Fishkin, and Beverly L. Harrison. Design and technology for Collaborage: collaborative collages of information on physical walls. In *12th annual ACM symposium on user interface software and technology*, pages 197–206, November 1999.
18. Helen Petrie, Valerie Johnson, Thomas Strothotte, Steffi Fritz, Rainer Michel, and Andreas Raab. Mobic: Designing a travel aid for blind and elderly people. *Journal of Navigation*, 49(1):45–52, 1996.
19. Nick Ryan, Jason Pascoe, and David Morse. Enhanced reality fieldwork: the context aware archaeological assistant. In *Computer Applications in Archaeology*, 1997.
20. Sue Long, Rob Kooper, Gregory D. Abowd, and Christopher G. Atkeson. Rapid prototyping of mobile context-aware applications: The cyberguide case study. In *Mobile Computing and Networking*, pages 97–107, 1996.
21. Keith Cheverst, Nigel Davies, Keith Mitchell, and Adrian Friday. Experiences of developing and deploying a context-aware tourist guide: the guide project. In *International Conference on Mobile Computing and Networking*, pages 20–31, 2000.
22. Staffan Bjork, Jussi Holopainen, Peter Ljungstrand, and Regan Mandryk (editors). Special issue on ubiquitous games. *Personal and Ubiquitous Computing*, 6:358–458, December 2002.
23. Robert Headon and Rupert Curwen. Movement awareness for ubiquitous game control. *Personal and Ubiquitous Computing*, 6:407–415, December 2002.
24. Staffan Bjork, Jussi Holopainen, Peter Ljungstrand, and Karl-Petter Akesson. Designing ubiquitous computing games - a report from a workshop exploring ubiquitous computing entertainment. *Personal and Ubiquitous Computing*, 6:443–458, December 2003.