

# **Location-Aware Asset Tagging Using Crickets**

Grayson Giovine  
Professor Seth Teller

May 16, 2005

## **Abstract:**

The goal of this project was to assist the MIT Property Office with the process of continuously tracking assets on campus. They do this by placing a barcode on any movable item purchased by an MIT department over \$1000, and maintain these “tagged” assets in a database. Originally, the idea was that we could attach a location-aware device (Cricket) to their barcode readers such that each time a scan was made, the position information for that item was recorded, in order to find it more easily the next time. However, in speaking with employees in the Property office about their biggest time strains, this issue did not seem nearly as important as the personal interaction needed to resolve unaccounted-for items. They would often spend months tracking down unaccounted-for items, and the process of doing so was extremely tedious. Therefore, in order to help streamline this process, I built a tool that can manage unaccounted-for items in a database, find the appropriate email addresses corresponding to each item, and email the responsible persons in an intuitive manner.

## **Introduction:**

The Global Positioning System is a technological advance that has supported many extremely useful functions. However, GPS has a drawback, and that is that it cannot work indoors. There are too many obstacles for the satellite signals to travel through. Therefore, if we want to achieve GPS-like functionality indoors, we need a new type of system. This system, which will be described, is the Cricket Indoor Location System.

Just as GPS is useful for many functions, the Cricket Indoor Location System is useful for many functions. One is that it can be used to assist with what is known as “asset-tagging”. This consists of tagging with a barcode all valuable items in a particular place. MIT does this, for example, for all “assets” costing over \$1000. In order to keep track of all these assets, they need to be periodically scanned and the records maintained in a database. Crickets could facilitate this process. This is because if a barcode scanner were interfaced with a Cricket, the building and room information would already be known, and this would speed up the asset-tagging process.

After much dialogue with employees in the MIT Property Office, however, this Cricket functionality was not as desirable as originally thought. There were three main reasons for this. First, due to the fact that no Cricket infrastructure exists in most areas on campus (just the 3<sup>rd</sup> and 9<sup>th</sup> floors of the Stata center), the functionality would not get utilized for the most part. Second, the Cricket device is large enough where it might become a bit of a hassle to need to carry it around along with the barcode reader, even if it was attached. Finally, the amount of work saved overall would be minimal. The person doing the scanning only needs to enter the building and room number manually in order to begin scanning, and needs to change these parameters occasionally. Though it would be nice to eliminate this manual step, the reality is that it only takes about 10 seconds to do, and therefore would not be a tremendous savings.

Despite this lack of enthusiasm for Cricket-functionality on their barcode readers, there was a major time strain the Property office conveyed to us - the personal interaction needed to resolve unaccounted-for items. It took a great deal of effort to type up the unaccounted-for items reports, find the email addresses pertaining to the responsible people for each item, and send these emails. Furthermore, the people that receive these emails are often too busy with classes and research to find items, and it will often take a number of reminder emails to get a response. The Property office might spend months tracking down some items, and this process was extremely tedious and time-consuming.

In order to help streamline this process, myself and another student, Tina Hsieh, built a set of tools that will help automatically manage the process of resolving unaccounted-for items. I built a tool that the Property office will use – it manages unaccounted-for items in a database, finds

the appropriate email addresses corresponding to each item, and emails those people. Tina wrote a tool for the other side – the users. Once a user receives an email from a member of the Property office, they can access Tina’s tool in order to resolve where the item is, and send an email back.

In this way, the tools work together to form an end-to-end system that both the Property office and general users can utilize to resolve unaccounted-for items quickly and efficiently.

## **Background:**

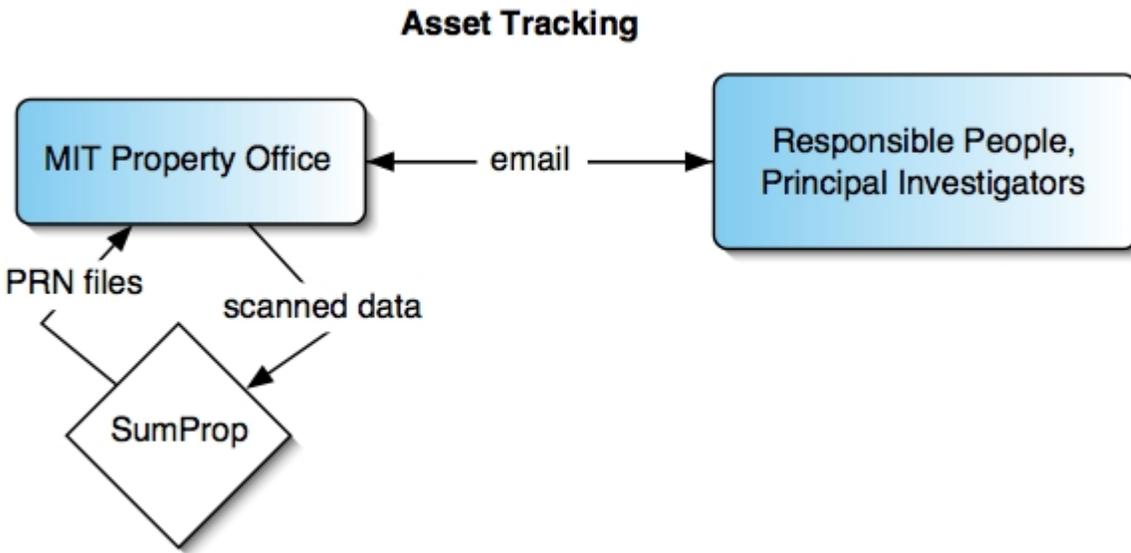
In order to understand this project, there are two key topics: asset-tracking and location-aware systems. Asset-tracking will be discussed on a level specific to MIT, as well as on a general level. In terms of location-aware systems, Cricket will be discussed, as well as alternatives to it.

### *MIT Asset-tracking*

The process of tagging and tracking assets at MIT occurs on a two-year cycle. There are a number of employees in the MIT Property Office that are responsible for going to buildings on and off-campus and scanning all items with MIT barcodes. Since the people that do this have done so for a number of years and are experienced with scanning, they know how to go about scanning the items in a particular building such that few items are missed, and that every room is scanned.

After each day’s scan, the scanner will go back to the office and upload the data to his local machine. He would place the barcode reader (the PDT 6800 by Symbol is the one used) in its cradle, which is connected to the computer and enables a file transfer from the reader. Once this is done, that text file can be uploaded to a database called SumProp, which contains all the tagged assets at MIT.

Periodically, a report will be generated using SumProp called a PRN file. This is an exception report, containing items that have not been found in the last specified amount of time (this is a parameter the person would set at the time he runs it). He would then print out this report, and manually go through each unaccounted-for item, find the names responsible for that item, look up their email addresses using the MIT directory, and send an email containing the relevant information. This information would need to be typed manually as well, copying it from the printed report.



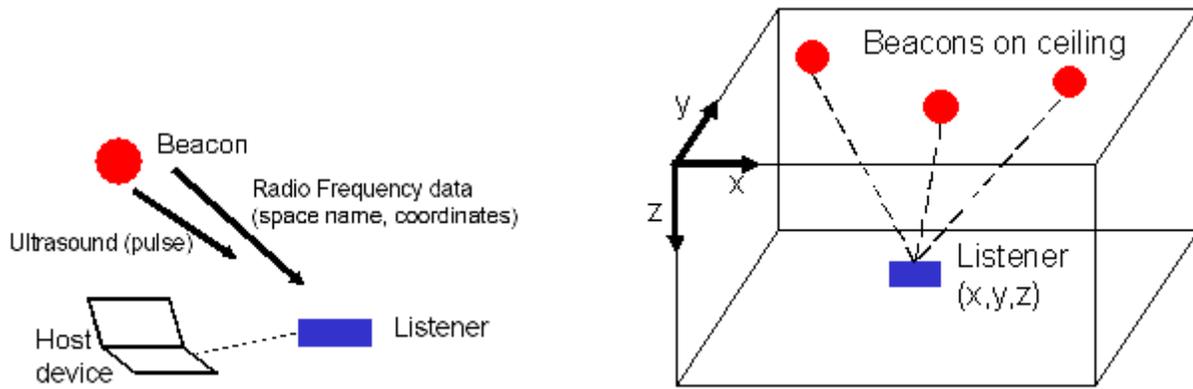
**Figure 1:** Asset-tracking by the MIT Property Office: after each day’s scan, the scanned data will be uploaded to the SumProp database. Periodically, SumProp will be used to generate PRN files, which are used to send emails to responsible people and principal investigators.

Clearly, this process was one that could be improved dramatically. This improvement will be discussed in the Approach section.

### *Crickets*

The way that the Cricket Indoor Location system works is relatively simple. It consists of three components – a beacon, a listener, and a host device. The beacon is the active component of the system. It sends out a periodic “chirp”, consisting of a radio frequency (RF) message and an ultrasound pulse. Since the RF message travels faster than the ultrasound pulse, the time lag between the two signals arriving at a listener can be used to determine the distance the listener is from the beacon. The host device is attached to the listener and processes the results.

If a listener can receive this information from three beacons in a room, the host device can use triangulation to determine the three dimensional coordinates of the listener.



**Figure 2:** Cricket system consists of three parts that work together to determine 3-D coordinates.

In order to do asset-tagging in buildings using Crickets, we can use the fact that the beacons broadcast information about their location to the listener. Therefore, if we have a Cricket listener attached to a barcode scanner, the Cricket already knows the important information about the asset as a person moves from room to room in a particular building. The barcode that is scanned can be combined with the building and room information in order to maintain the asset database.

### *Location Systems*

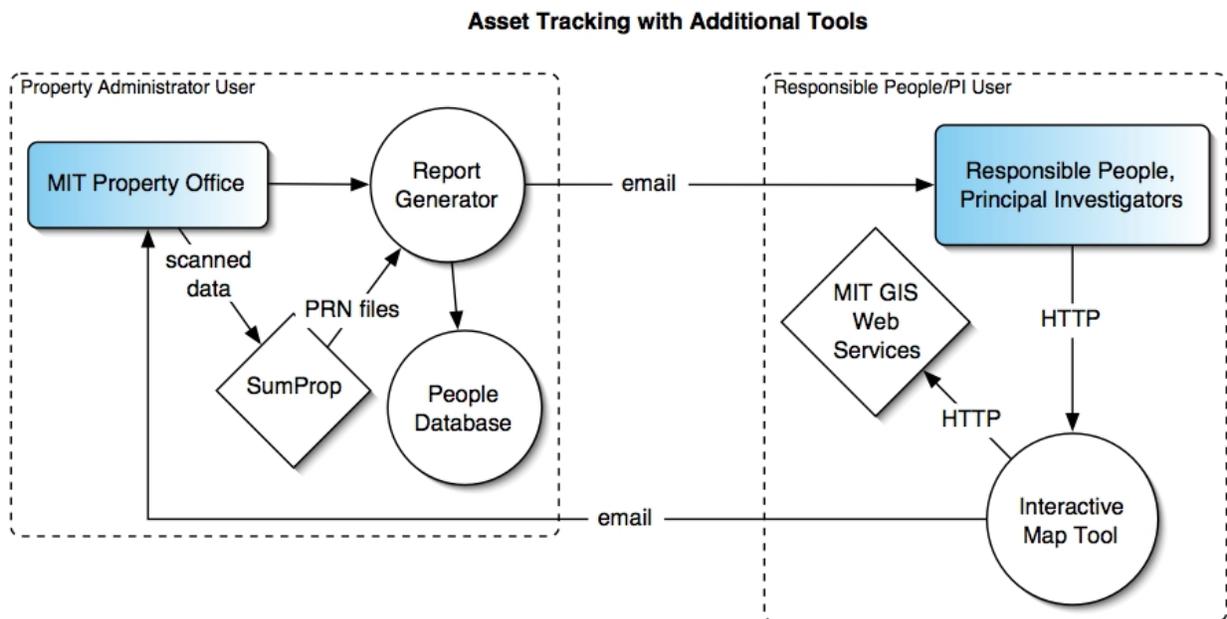
There have been similar systems in place that perform similar functions to the Cricket Indoor Location System. Most of them have a similar architecture to that of the Cricket system, where some device sends out a signal and another listens for it in order to determine position. There is one key variation in many of the other systems, however. This variation is that the beacons are passive rather than active. Therefore, there is an active component that a person might carry around. This component would send signals to the passive beacons in order to determine position.

### *Other Types of Asset-tagging*

There have been other ideas for automating asset-tagging besides using an indoor location system. One such example was to use RFID tags instead of barcode tags in order to better keep track of items. This is better in terms of dynamically maintaining a record of where each asset is. With the Cricket asset-tagging, if a particular asset is moved, it would have to be re-scanned in order to determine its new location. With the RFID system, this could be dynamically adjusted because of the signal coming from the asset's RFID tag. Obviously there are drawbacks to this system - security and privacy are two of the major ones.

## Approach:

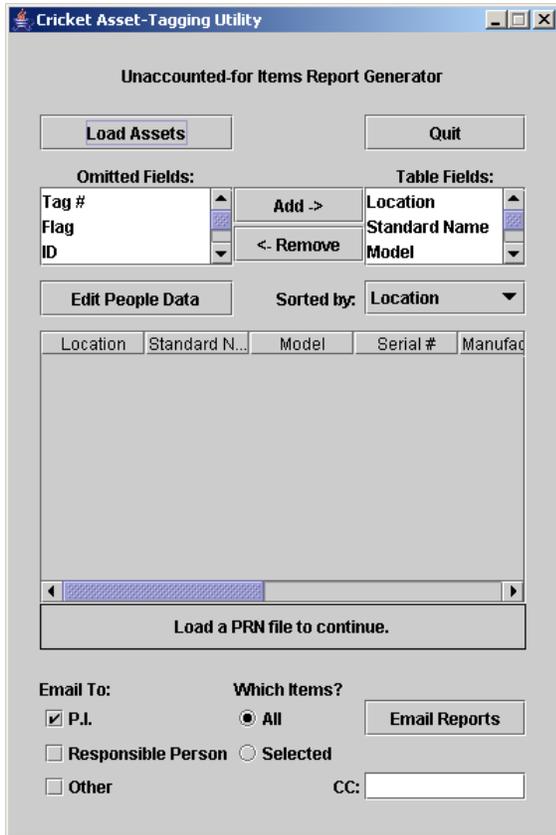
In order to speed up and automate the interaction between the property office and the people responsible for unaccounted-for items, we added a layer in between the two. There are two main modules. One module contains the utilities used by the MIT Property Office. This includes their SumProp database, as well as the report generator and the people database. The report generator accesses the people database in order to obtain email addresses. The second module is for the responsible people. Once they receive an email from the Property office, they go to the interactive map tool, and from there can load their unaccounted-for items (using a confirmation number contained in the email they were sent). If they know where the items are, they can be filled in using this tool, and the updates are sent to the Property office.



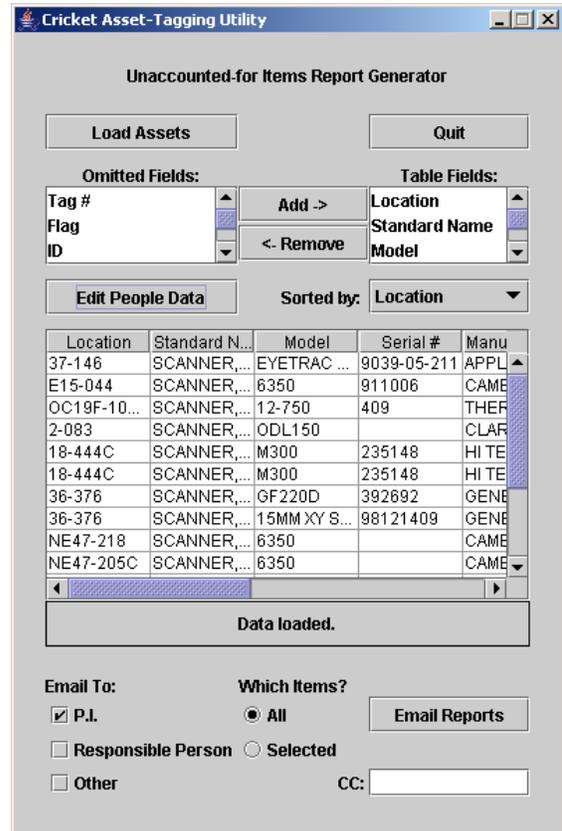
**Figure 3:** Flow chart of the end-to-end system. The Property administrator uses the report generator to send emails to the responsible people, and they use the interactive map tool to send emails back to the Property office.

## Report Generator

The report generator is a Java application that takes in a PRN file generated from the SumProp database, parses through it to separate each item, and allows the user to send emails to each responsible person.



(a)



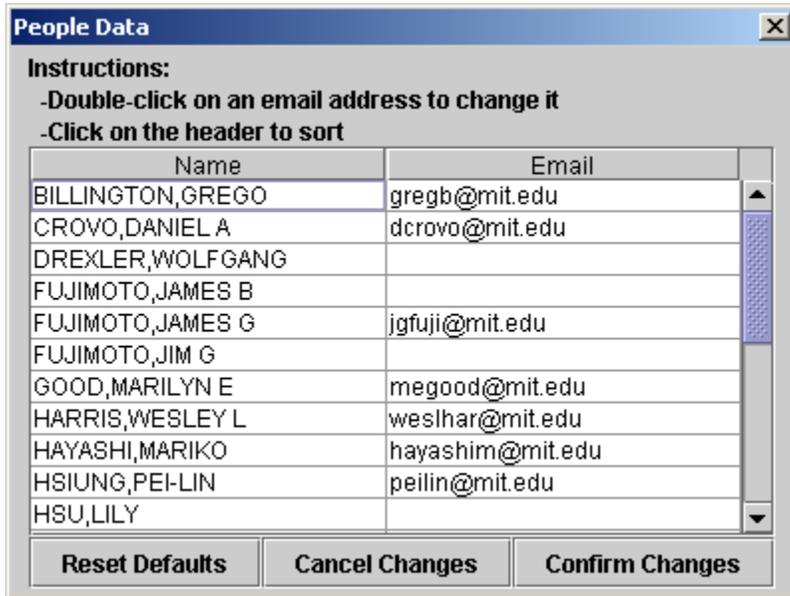
(b)

**Figure 4:** The Report Generator user interface: the applet on the left is the one that appears when the program is first run. Once a report is loaded, using a file chooser, that appears in the table.

When the application is first loaded, the user is asked to load the PRN file. Before doing that, attempting to use the other features of the application (besides quit), will prompt the user to first load the PRN file.

After that happens, it is parsed and placed in the table, as shown in Figure 4b. From here, the user can modify the table as he chooses, shifting columns to be in any order, adding or removing columns (there is a default set of columns which are the most frequently used), and sorting. Sorting can occur either by using the drop-down box at the top of the table (it adds or removes fields based on the columns in the table), or by clicking on the table headers.

Once the PRN file is loaded, the application finds all the email addresses for the names listed as principal investigators and responsible persons. It uses the people database, which is maintained locally on the machine where the program is run.



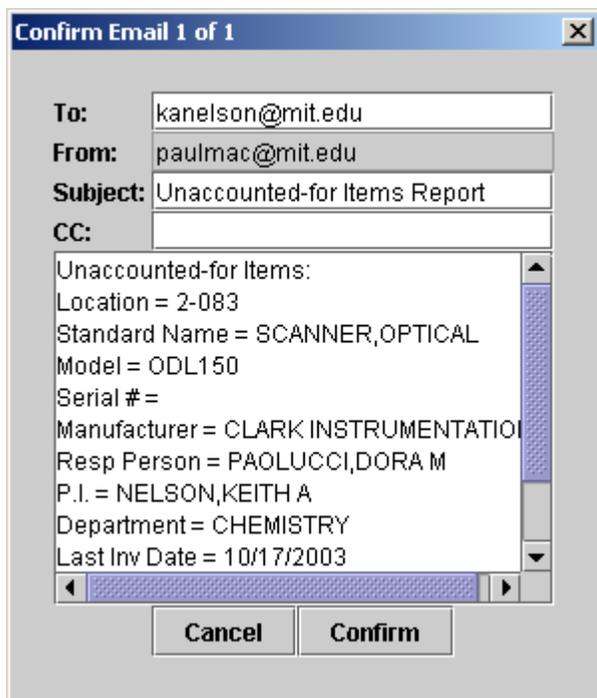
**Figure 5:** The PeopleView dialog: when the user clicks “Edit People Data” on the main interface, this dialog will appear. It allows the user to change the default email addresses for the users in the particular PRN file.

The database contains two tables, “export” and “dynamic”. The export table contains the information for everyone in the MIT people directory. Eventually, we would like the program to directly access the MIT directory, but for now it accesses a dump of the directory into a local MySQL database. The dynamic table is a table used to override the email address for a particular name. Therefore, if in Figure 5 we wanted to send items for Greg Billington to email address [ggiovine@mit.edu](mailto:ggiovine@mit.edu), rather than the original [gregb@mit.edu](mailto:gregb@mit.edu), then that could be changed and saved. This would place the changed email address and corresponding name in the dynamic table.

When the application looks up the names, it first looks in the dynamic table. If a particular name and email address is there, it will use that. If not, it will default to the original email address in the export table.

Using the table of assets and the email address corresponding to each name, the user can send emails. The user can select one or more items to send, or send all the items in the list. He can send to the PI, Responsible Person, and Other (one or more of those options). In addition, he can CC himself or someone else, by typing in the bottom right-hand corner of the main interface.

Once the “Email Reports” button is pushed, the items to be sent to each person are grouped (such that if 4 items were going to the same person, it would be in the same email instead of 4 different ones), and confirmation dialogs will pop up. One such dialog is shown below.



**Figure 6:** The PeopleView dialog: when the user clicks “Edit People Data” on the main interface, this dialog will appear. It allows the user to change the default email addresses for the users in the particular PRN file.

Each dialog contains a To, From, Subject, CC, and Body field, just like a typical email client. Everything on this dialog is editable except for the From field, since that is tuned to the particular user using the application (in this case it is Paul MacDonald in the Property Office). More email addresses can be added to the To or CC fields with a comma. The user can also add comments at the top of the email, in case he wanted to say something specific to that particular person.

Once the email is confirmed, a number of up to 6 digits is generated. This is the confirmation number for that email. This is added to the body of the email as follows:

Your confirmation number is: 809703

Unaccounted-for Items:  
Location = 2-083  
Standard Name = SCANNER,OPTICAL  
Model = ODL150  
Serial # =  
Manufacturer = CLARK INSTRUMENTATION  
Resp Person = PAOLUCCI,DORA M  
P.I. = NELSON,KEITH A  
Department = CHEMISTRY

Last Inv Date = 10/17/2003  
Amount = \$5930

**Figure 7:** The body of an email, with the confirmation number added at the top.

The email is then sent using the java.mail package. The emailing code uses mail server pacific-carrier-annex.mit.edu for any email addresses that end in “@mit.edu”. This is because that mail server has username checking, and therefore can reject an email being sent to an invalid username. If the email address does not end in “@mit.edu”, then it will send on outgoing.mit.edu, which does not have any checking. Sending on this mail server allows the program to send to any address, but the drawback is not being able to tell if the address is valid.

Finally, once the email has been sent successfully, a record of it is added to the “sentemails” table in the comm database. This contains the email address, the tag number (one entry for each item in the email), the Count (used for determining whether a particular email is a second or third notice), and the relevant fields for that item (location, model, serial #, etc).

### **Interactive Map Tool**

Once the emails are sent to the responsible people, the Interactive Map tool must be used. It is a Java web applet that displays an interactive view of the floor plan where the user’s unaccounted item were last seen. The application uses MIT GIS Web Services to obtain svg files of the campus.

When the program is first started, the user is prompted for a confirmation number. This number was in the email they received. The number they type will be sent in a query to the comm database (shared with the report generator).

Once the query is made and returns with the items for that number (there may be more than one), this will be formatted as an xml file.

```
<?xml version="1.0" encoding="UTF-8"?>
<itemlist>

<item name="SCANNER,OPTICAL" tag_num="0282648">
<location>2-083</location>
<responsible>PAOLUCCI,DORA M</responsible>
<PI>NELSON,KEITH A</PI>
<dept>CHEMISTRY</dept>
<man model="ODL150" serial="">CLARK INSTRUMENTATION</man>
<lastinv>Fri Oct 17 00:00:00 EDT 2003</lastinv>
</item>

</itemlist>
```

**Figure 8:** XML-formatted string of the relevant data.

This information will be parsed and the information drawn in a sidebar for the user. The user can then edit this information by either changing the text, or by drawing, circling, and writing on the map. Once these updates are made, the user can send it to the Property Office. This email will contain an attachment of a JPG file containing a view of where the item is.

Once this email is sent, it goes to the Property administrator (Paul MacDonald). It was decided this intermediate step was necessary such that users are not directly editing the SumProp database. The administrator can then either accept the location of the item and upload it to SumProp, or go to find and scan the item at the location where it supposedly is.

End-to-end, this system works well, and although it does require some manual input, these steps are necessary.

## **User Comments:**

After completing each version of the program, it was sent to a member of the Property office for evaluation. Through a continuous feedback loop, the program was improved and features were added to help meet the needs of the Property office.

Paul MacDonald made the following comments about the most recent version of the report generator:

- 1) Eliminates the step of manually typing pertinent information for each item by supplying all necessary fields from the captured prn file.
- 2) Helps to eliminate the added step of searching the online directory for the email addresses of recipients.
- 3) Very user friendly, with a screen that is clear, concise and easy to follow. (although a tutorial for the uninitiated would be needed)
- 4) Very flexible - all field options as well as all email options are easily added or subtracted to suit specific needs.
- 5) Presents the information within the email in an organized and professional manner.

Based on these comments, the tool certainly seems to be a desirable one, and once a few more modifications are made, it will very likely be used in the daily work of the Property office.

## **Implications:**

Though the scale of the implications for these tools is fairly small, it will have an enormous impact on the way the Property office follows up on unaccounted-for items. Typically, it would take an entire day to send out emails to responsible people in an average-size PRN file. Over the course of the entire 2-year cycle, this is a significant chunk of time. The report generator reduces this to a matter of minutes, so it will be a significant savings. Furthermore, the interactive map tool is good because it provides an easy-to-use, streamlined way for people to respond with the locations of unaccounted-for items. Due to the significant time savings this set of tools provides, the MIT Property Office will surely use it in the future.

## **Future work:**

There is much future work to be done with this project. This work will largely be completed during the summer. First, the database used to communicate between the report generator and interactive map tool needs to be placed online in a secure manner. Though the data is not particularly interesting, we do not want this viewable by anyone that is unauthorized. Furthermore, instead of using a local database containing the information about people in the MIT directory, we would like to be able to directly access it.

Also, instead of using confirmation numbers, the email sent should contain a link unique to that user, which can be accessed and essentially fills in the confirmation number without a prompt.

In the future, Cricket functionality can be added as well. However, before being able to do this, we need to establish a permanent infrastructure of Cricket devices. This can be small-scale at first (i.e. the 3<sup>rd</sup> floor of Stata Center), and can eventually expand. We need to show asset-tagging is feasible on a small-scale with Crickets, using this small permanent infrastructure. Once that is accomplished, the scope can become larger, and the asset-tagging can expand along with the deployment of Crickets.

## **Conclusions:**

This project succeeded in that it dramatically reduced the amount of work the MIT Property office needed to do. There are a few manual steps needed along the way – for example, neither program interacts directly with the SumProp database, because the Property office wanted to have a human layer between the two. However, the process is largely automated, and speeds up the resolution process for items by orders of magnitude.

## References:

jGuru: Fundamentals of the JavaMail API:

<http://java.sun.com/developer/onlineTraining/JavaMail/contents.html>

MIT Controller's Accounting Office: <http://web.mit.edu/cao/www/>

Nissanka B. Priyantha, Anit Chakraborty, Hari Balakrishnan. *The Cricket Location-Support System*, Proc. 6th ACM MOBICOM, Boston, MA, August 2000.

O'Hanley, John. "Send an Email." <http://www.javapractices.com/Topic144.cjp>