

RFID Event Specification Using Templates in Scenic and Event Notification

Kayla Gould, Evan Welbourne, Magdalena Balazinska

Department of Computer Science and Engineering
University of Washington, Seattle
Seattle, WA 98195

{keucken, evan, magda}@cs.washington.edu

ABSTRACT

Scenic is a program that allows the user to graphically specify an event to be detected by the RFID Ecosystem in the Paul G. Allen Center at the University of Washington. The user drags icons representing people, places, items and “event primitives” [5]. In this paper we discuss how Scenic has been extended to (1) save and recall an event definition, (2) edit previously created events, and (3) create event templates.

Notifier is a new program we created that allows the user to subscribe to notification for an event that they have created using Scenic. The user enters contact information such as an email or a phone number and specifies which method to use to be notified when the event occurs. The system monitors the readings from the RFID Ecosystem and when it has detected that the event has occurred, it sends out a notification via the specified medium.

Categories and Subject Descriptors

H.4 [Information Storage and Retrieval]: Miscellaneous; H.5 [Information Interfaces and Presentation]: Graphical user interfaces (GUI)

General Terms

Design

Keywords

RFID

1. INTRODUCTION

There are two specific goals set out by this project. The first goal is to enable users to more easily and efficiently create events by updating previously created events and by creating and updating event templates. The second goal is to build a working notification program to allow users to subscribe to and receive notifications about RFID events that occur in the project’s environment.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Some Conference '09, Some Date, 2009, Some Location.
Copyright 2008 ACM 978-1-60558-139-2/08/06 ...\$5.00.

Another more general purpose of this project is to explore the usability of programs designed to allow average, non-technical users to manage RFID events in a pervasive RFID environment. The project is designed to make use of the RFID Ecosystem set up in the Paul G. Allen Center at the University of Washington [4]. The ecosystem consists of a network of RFID antennas and readers that have been strategically placed on all six floors of the building and the basement. To make use of this ecosystem, users volunteer to wear RFID tags where-ever they go and also usually tag some of their personal items such as books, keys, laptops, and backpacks.

The programs developed under this project will allow users to track occurrences of interest involving themselves, their tagged items, and places that are covered by the RFID antennas. These occurrences are called “events” and may be such things as when a person and their advisor meet in the advisor’s office, when a person is seen leaving the building without their keys, or when a stationary inanimate object changes location (*i.e.* is stolen). When events such as these happen, the system will send out notifications to let the users know that events they care about are happening.

1.1 Motivation

Humans are often forgetful. We use many different ways of reminding ourselves of important details, such as post-it notes, emails to ourselves, and entries on calendars. Most of these methods are created and then must be visible at the critical time for the reminder to be effective. For example, consider the case where you write a post-it note reminder to bring a book to a meeting to give to someone and must see the note as you are leaving your office. The reminder may be overlooked or may not be necessary if you already have the book in your hands.

Our system will allow the user to subscribe to conditional reminders which, depending on the notification method selected, may be able to deliver the reminder right to you no matter where you are.

This project extends and integrates with the Cascadia System [5]. Cascadia is a middleware for developing applications that need to specify, detect, and manage RFID events. It allows developers to incorporate RFID events without having to deal with the inherent uncertainty and ambiguity in RFID events. Cascadia also handles event detection, thus allowing the developer to merely subscribe to an event and wait for messages saying that the event has happened.

1.2 Related Work

Our project is similar to the SPECS project done by Mik Lamming [2] where they used small peer-to-peer computers mounted on people and objects to keep track of which items were with a person so that the person can be reminded when they leave an item behind. Their system required a very minimal number of devices to work but was only designed for keeping track of items. Our system can be told to remind you if you forget certain items but events other than forgetting things can be tracked, where the SPECS project was only scripted to keep track of items. We also use a building-wide RFID ecosystem of readers instead of peer-to-peer computers.

The RFID reminding project done by Boriello *et al.* [1], worked on keeping track of tagged inanimate objects. Their system allowed the user to write reminders about what items they should have with them at certain times. The system would watch for those items and then remind the user on a special wristwatch when they forgot an item. Their method of specifying “reminders” was text-based and might make an average user disinclined to learn the system, where our method of specifying events, which can also be used to keep track of items, is graphic and perhaps much more comfortable for an inexperienced user.

Sohn *et al.* [3] worked on a location-based mobile reminder system called Place-Its. Place-Its allows a user to go to and name places and then create reminders to be deployed when they are once again near that place. They found that location-based reminders are useful even when the accuracy of the location detection is not very good.

1.3 Contributions

This project builds on existing research but makes the following contributions.

1. **Abstracting from technical details.** Unlike programs such as the reminder system by Boriello or the SPECS project [1, 2] users of our program will be able to specify what to be notified about without needing to learn a scripting or programming language. While a background knowledge of how the system works is helpful, as long as the user can drag and drop icons and use right-click menus, they should be able to create event specifications using Scenic. The events can also be managed very easily with the templates and tables that have been added to Scenic (Section 2).
2. **Allowing users to choose notification details.** Many of the projects previously developed allow the user to only use one method of notification such as a mobile phone, or a wristwatch [1, 2, 3]. Our program allows the user more choices of notification and even allows them to customize the notification message (Section 3)

We present the event specification ability of Scenic in Section 2 and the notification ability of Notifier in Section 3. We discuss the project’s current status in Section 4 and lay out our planned user study in Section ???. Finally, we present related work in Section 1.2, and conclude in Section ???.

2. EVENT SPECIFICATION

On the base level, the data obtained from the RFID Ecosystem is a series of RFID tag reads by particular antennas. To the average user, this level of data is not very useful. Our goal is to abstract the data, attach meanings to it, and

allow management of it in such a way that a non-technical user could find it helpful in their daily life.

One such management abstraction is the ability of the user to specify RFID events. An RFID event is a set or series of sets of tag reads that have some meaning associated with them. One simple event is the sighting of the person tag for Person A and the person tag for Person B by the same antenna at the same time. Such an event could be classified as a meeting or encounter, depending on the length of time that Persons A and B were seen together.

In our experiment, events will be specified using a program called Scenic [5]. The program provides a visual way for the user to specify scenes of an event. The user can, for example, add a scene to the event, drag down icons that represent a person, the modifier “Outside,” and an icon that represents a place. They could then right-click the icons to specify that the person is themselves and the place is the Allen Center. They could then add a similar scene consisting of themselves inside the Allen Center. The combined scenes form an event where the user is seen entering the building. While this is relatively easy, the accuracy is sometimes questioned because the relative positions of the icons can change the meaning of the event.

This project has added the ability to specify events using previously created templates. These will be created by developers and the user will have the option to create their own template if they wish. A template will consist of an event, such as the building entering event above, but with some of the details left out. The user will supply the missing details, enter a name for the event, and save it.

We anticipate that the incorporation of templates will help increase the confidence with which people can specify events in Scenic. It may also increase the accuracy of the events specified, as measured by whether or not the event actually describes what the user intended.

3. EVENT NOTIFICATION

In addition to providing Scenic for event specification, our project provides a program called Notifier that allows the user to subscribe to notification for an event that they have previously specified. The user will be able to view a list of events that they can subscribe to, select one of the events, click the Subscribe button and then enter details about how they wish to be notified.

The user will be able to choose whether to be notified by email, SMS text message, or phone message. They will be able to customize the message that is sent when the event happens and in the case of the phone notification, they will be able to choose whether to use a text-to-speech notification or record their own message.

There will also be an option to change the probability threshold for the notification. For example, if the system is only 60% certain that the event has occurred, it will not bother the user unless they have specified a certainty threshold of 60% or less. This will allow the user to specify a low threshold for events that are more critical to detect, such as an inanimate object moving from a predetermined spot (possibly being stolen), and to specify a high threshold for events that are less critical and they don’t want to be bothered about unless the system is very certain.

For the actual event detection, the system will make use of the Cascadia Event Manager which will take the event specified by the user and notify the project’s system once

it has decided that one of the user's events has occurred. Cascadia uses a particle filter to take the raw data from the antennas, smooth it, and assign probabilities to where the tag was located. The smoothed data is then sent to PEEEX, a probabilistic event detector, which monitors for data that matches the specifics of an event.

Once the system has ascertained that the probability that a specific event has occurred exceeds the user's threshold probability, it will send the user some sort of notification. This will entail sending out an email, sending a SMS text message, or calling the user's cell phone and playing a pre-recorded message when they pick up.

4. PROJECT STATUS

A lot of progress has been made toward completion of the project. Scenic is nearing the point of being really usable. The visible portion of Notifier is functional, however the portion that actually sends out the notifications is barely started.

4.1 Scenic

The GUI of Scenic has been extended quite a bit. The user can create a template which is saved as a name, a description, and an XML representation. They can also create new events either from scratch or from a template that they or a developer has previously specified. GUI components have been added to allow the display of previously specified events and templates so that they can be viewed, edited, and deleted.

Currently, events and templates cannot be loaded visually but the XML can be retrieved from the database and will soon be able to be loaded into Scenic as scenes. Scenic has also been modified so that the items that the user can create events with are real tagged items instead of fake, hard-coded item names.

The list of single item names is currently the only list that is populated dynamically. The list of people and places will also be populated dynamically before the project is considered complete. People groups and item groups will probably be temporarily removed since there is currently no source of data for those categories. The names of people will probably be populated from an existing table of RFID system users although ideally the names would come from some database containing people linked as friends. This would increase individual's privacy as it would prevent people from creating events with people they do not have permission to track. Until a meaningful database exists, place names will probably come from a fake table with manually populated data. It is anticipated that eventually another program will be completed that will be responsible for populating the Place table.

4.2 Notifier

For the Notifier application, a GUI has been created that will allow the user to subscribe to notification for an event that they previously created. The program prompts first-time users for their email and phone number. Currently the program prompts all users for contact information because the system is not hosted in a way such that the user can actually log in, and so for development purposes does not require contact information or store the answers.

For returning users, they will automatically be forwarded to a list of events that they are currently subscribed to notifi-

cation for. From that list the user can choose a notification subscription to modify, activate or deactivate a subscription, and delete a subscription. When an event is selected for modification, the details of the event are loaded and the user may make changes, including changing the notification details. Activating or deactivating a subscription currently only changes the value of a single field in the database but once the program is actually connected to the event detection system it will change whether or not the event is watched for. The difference between deactivating and deleting a subscription is that a deactivated subscription remains in the database while a subscription that is deleted is deleted from the database and must be recreated if the user wants to use it again.

In addition to viewing currently subscribed to events, the user can select an event from a table of events previously created using Scenic and click a button to subscribe to notification for it. They can select the notification type of email, SMS, or phone and customize the message except in the option of Phone:Record your own). For events that the user specifies email or SMS as the notification type, their default or customized message is stored. The user can also customize and preview a text-to-speech phone message, but the phone messages do not store yet and you can't yet record your own message.

The back end of Notifier has barely been started. It will consist of a servlet or basic application that will reside on the server and will listen constantly for messages from the Cascadia event manager that say that an event has occurred. When an event detection message comes in, the program will look up the details of the event that has occurred and carry out the corresponding notification. This will entail looking up the email or phone number of the person who owns the event and either sending out an email, a text message, or calling up the person, waiting until they answer their phone, and playing the prerecorded message.

4.3 Challenges

The biggest challenge so far has been figuring out how to use the Google Web Toolkit (GWT) for the project. GWT is fairly new and not terribly well documented. While in the future it may be a very good platform for development, it is still fairly basic and often many classes must be created to extend the base classes into what you need.

For Scenic, it has been difficult to figure out how to adapt and add on to a program that someone else created and currently has 90 classes. Functionality also had to be added to the base classes of Scenic to allow the events to be stored in an intermediate XML format and correspondingly to be recreated from the XML.

For Notifier, the hardest part has been getting the notifications to work. There are java APIs for email and text-to-speech but they are complicated and the documentation does not seem to include information on how to do what is needed for this project.

Despite the challenges mentioned above, the project is proceeding successfully and is expected to be completed at least in some simple form by the end of the allotted time for the project.

5. EVALUATION

The evaluation for this project will be somewhat incomplete since the required supporting RFID Ecosystem pro-

grams are not all in states of completion such that they can be used by real users. However, the evaluation is expected to still be very useful.

5.1 User Study

The criteria for evaluating Scenic will consist of how easy it is for users to use the interface. If resources permit, two tests will be run. One test will have a group of control users create specific events from scratch. The other test will have users create the same specific event from templates. The amount of time that the users take to make the event will be recorded, as well as their evaluation of how easy it was to create the event. The times and ease-of-use ratings for events specified with and without templates can then be compared to see if using templates increases the ease with which the users can specify events.

The criteria for evaluating Notifier will be somewhat more subjective. Ideally, the same users as in the Scenic test groups will be asked to subscribe to notification for the events that they have created. The results of the test will come in the form of questionnaires or interviews with the users to see how easy they thought it was to use the program and whether or not they would be interested in using a similar system to keep track of RFID events in their lives.

5.2 Technical Performance

Evaluation of the actual notification methods will have to be a user-less laboratory study due to the reasons mentioned previously that keep us from being able to use the real Cascadia event detection system. Fake event detection messages will have to be generated to test that the system actually sends the appropriate messages when events are “detected”.

The system will also undergo a variety of benchmark testing to determine how fast the system works. In the case of Scenic, this will include timing how long it takes to save an event or template, load an event or template, and refresh the tables of events and templates. In the case of Notifier, this will include timing how long it takes for the system to send out the notification once the event has been detected.

6. REFERENCES

- [1] G. Borriello, W. Brunette, M. Hall, C. Hartung, and C. Tangney. Reminding about tagged objects using passive rfids. In *UbiComp 2004*, 2004.
- [2] M. Lamming and D. Bohm. Specs: Another approach to human context and activity sensing research, using tiny peer-to-peer wireless computers. In *UbiComp 2003*, 2003.
- [3] T. Sohn, K. Li, G. Lee, I. Smith, J. Scott, and W. Griswold. Place-its: A study of location-based reminders on mobile phones. In *UbiComp 2005*, 2005.
- [4] University of Washington. RFID Ecosystem. <http://rfid.cs.washington.edu/>.
- [5] E. Welbourne, N. Khoussainova, J. Letchner, Y. Li, M. Balazinska, G. Borriello, and D. Suci. Cascadia: A system for specifying, detecting, and managing rfid events. In *MobiSys 2008*, 2008.