Distributed To-Do Lists Using Online Social Networks (Progress report #2)

Clifton Forlines, Justin Ho, and Koji Yatani
forlines@merl.com, {jho, koji}@dgp.toronto.edu
Department of Computer Science, University of Toronto

ABSTRACT

Shopping is a common activity in daily life. Not only do people typically visit multiple locations to purchase their goods, but the realization of a need to purchase a particular item may come at inopportune times. Most people carry scraps of paper for these moments to jot down reminders for themselves to purchase particular items. Furthermore, the typical household common area may have a public to-do list, where occupants may scribble down items that need to be purchased on the next trip to the grocery store. Leveraging the opportunistic reminder that an electronic post-it note affords, we observe the potential utility of a system that not only allows multiple people to input to-do items, but reminds participating users at opportunistic times of items to purchase. In this paper, we describe a system that leverages online social network relationships to allow users entering a coffee shop to purchase drinks requested by their friends. By avoiding a two-phase commit protocol, and by providing users with additional contextual information, we hope to minimize the total travel time and the total number of redundant trips necessary, while allowing for a time-optimal solution. We discuss technical challenges of the proposed system, and the remaining issues that still need to be studied.

INTRODUCTION

The average American spends over 6 hours a week shopping, with much of this time spent traveling from one location to another. In addition to requiring a long amount of time (our most precious resource), this traveling consumes great amounts of fuel at a large economic and environmental cost.

In this paper, we present a prototype system that allows friends in an online social network to share “to do” lists so that they can coordinate their shopping efforts in the real world. Our system ties a social network’s central “to do” list to location based services so that group members learn when they are in a position to help out a neighbor in the network. By simply entering a location to which a friend is planning on visiting at a later date, members are notified via their cell phones that they are in a position to help a friend out by purchasing them an item.

Our architecture avoids a distributed locking algorithm, which is likely to be unreliable and slow in our vastly distributed network. Instead, our users participate in a novel distributed, two-phase commit system to ensure that to-do items are purchased in an efficient manner. The awareness of the status of other co-located users facilitates the coordination among them to complete tasks on the shared to-do list without duplication of effort. It is our hope that the widespread use of such a system can save its users an untold number of hours of traveling between shopping locations; traveling that is redundant when one’s social network is viewed as a whole.

MOTIVATING SCENARIO

The following scenario provides a high-level description of our system and motivates the need for a distributed to-do list by showing the system’s positive effects on the total amount of effort spent by a small group of friends.

The Status Quo: An Inefficient Shopping Behavior

During her morning coffee break, **Alice** remembers that she needs to buy bread and dog food on her way home from work that day. Unknown to Alice, her co-worker **Bob** makes a quick trip during his lunch break to the same supermarket that **Alice** would drive to at 5pm. After picking up bread, **Alice** hops in her car and continues to the pet store. While in the pet store, **Alice** runs into her neighbor **Carol**, who is there buying cat food. They then drive home in separate cars.

Although their purchases of food can be completed by other people, they cannot coordinate their shopping well since they have no clue of the friends’ context. We thus believe that users can leverage their opportunities to ask their friend to purchase what they want by knowing the friends’ context. In the next section, we will see how the shopping behavior will change with our proposed system.

A Distributed To-do List

During her morning coffee break, **Alice** remembers that she needs to buy bread and dog food on her way home from work that day. She adds these two items to her online social network enabled “to do” list. After lunch, **Bob** knocks on **Alice’s** door and hands her the loaf of bread that he bought for her. He had gone shopping over his lunch break. On her way home, Alice stops at the pet store to buy dog food, and when she steps through the store door, her cell phone informs her that her neighbor **Carol** is in need of some food for her cat. **Alice** checks this item off of **Carol’s** list, purchases both the dog food for herself and the cat food for **Carol**, and heads back home for the evening.

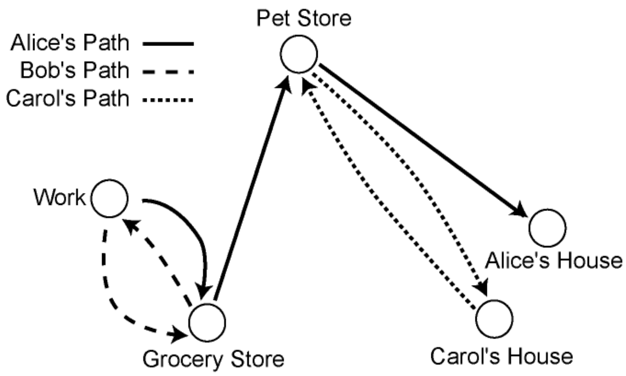


Figure 1. Many items on people's to-do lists can be easily purchased by friends in one's social network, if only friends were made aware that they were in a position to help.

In this scenario, the total number of store visits has been reduced from 4 to 2, and the total number of car trips from 3 to 2. Although we just see the case with a small number of friends, we expect that the users could coordinate their shopping more effectively with a large social network.

RELATED WORK

Location-aware Mobile Service

A series of papers have looked at replacing the post-it note with its electronic equivalent [2]. By adding additional sensing functionality, it is hoped that these electronic replacements will be able to leverage the opportunistic reminder - exploiting particular opportunities that the paper equivalent does not.

Sohn et al. describe the importance of using a ubiquitous device that users are already carrying, and pick the cell phone as the desired device [7]. This does not require users to carry additional equipment; the user tracking and notification functionality is augmented into an existing device that the user is carrying. Only by doing this in the users' natural setting, with no physical boundaries, they argue, can the contextual reminders be properly understood.

Ludford et al. added an enhanced interface to this system, allowing users to add or remove reminders at their desktop or mobile computers, and not just at their phones [3]. This made the post-it note truly ubiquitous; anywhere they have internet access, users are able to update their to-do list, and no longer restricted users to use the small display and input interface on their phones to do so. While some have looked at sharing read-only data to multiple users in this context [6], one of the open problems cited was the need to manage simultaneous updates with shared to-do lists and ensure that multiple people do not buy the same list item [3].

To solve this problem, we propose a basic framework to permit users to negotiate the completion of shared tasks. Users are able to broadcast an Estimated Time to Completion (ETC) if they were to complete the task. By examining other users' ETCs, the collective whole is able to come to an agreement about who should purchase a specific item to globally optimize the purchase time and effort, whilst preventing accidental duplicate purchases.

Integration of an Online Social Network into the Real World

The rapid uptake of online social network technology by the general population has resulted in the development of many services that offer to bridge that "gap" between the virtual and "real" world environments. One such service is Dodgeball [1]. Dodgeball notifies the location of a user to his/her friends with text messaging on a mobile device and supports their opportunistic social interaction in the real world. Another service is Meetup, which attempts to help build social networks by connecting users to other uses with similar interests by organizing social events collaboratively [4].

Our system leverages the opportunities around our shopping behaviours. These trips that we make are not necessarily planned or scheduled, but our particular location and context opens opportunities for us to purchase other items that our friends may also need. In our model, users are not explicitly tracked as they enter specific venues, and may decline to make specific purchases, if they decide that a particular relationship is unsuitable for the purchase that a friend has requested. Users are not explicitly notified that a particular friend has declined to make a purchase, but only notified if a purchase has been made by a friend. By keeping the purchaser in control of the exchange, we believe that users will use this service, even if monetary exchange needs to happen.

DESIGN PROBLEMS

Our design discussions led to an architecture with 3 major components: Cell phones, Beacons and the Social Network Server (SNS) as shown Figure 2. Cell phones and Beacons are connected over Bluetooth, and Beacons and Social Network Server communicate with each other through the Internet. The functions for Cell phones and Beacons are implemented in Java, and SNS is implemented in PHP and MySQL. Before explaining how our system works, we present several challenges in the implementation of our proposed system and discuss the solutions to them in this section.

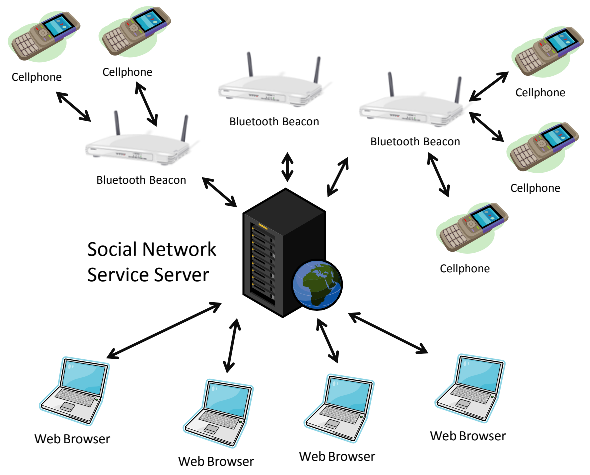


Figure 2. System Architecture

(Problem 1) Beacon should check whether Cell phones are in range constantly to know whether a user is still interested in completing a to-do.

Beacon should be aware of the context of each Cell phones. For example, Beacon should know whether a user has left the place before completing a to-do since the user may leave without notifying he/she is giving up the completion of the to-do. Therefore, once Cell phone is out of range for a pre-defined time, the system regards that the user has left the place and Beacon updates the status of the request. We have a threshold of time to determine the user's leave. Beacon may lose the connection temporarily in a noisy surrounding, but the connection will be recovered in a short time. In this case, the system considers that the user is still interested in completing the to-do.

(Problem 2) Beacons are supposed to be connected to public network connections and may be unable to accept inbound connections from SNS due to the firewall.

Beacons must initiate all communications to SNS and keep the entire system consistent. Beacons must be aware of all active Cell phones in the area, and any active requests that have been broadcasted to the Cell phones. Therefore, we decided to make Beacons send HTTP GET/POST requests to SNS proactively to keep being updated.

(Problem 3) Users can be invisible from the system whenever they want.

Users may not want to help their friends' purchase because they are in hurry. They may not want to let the system know their location in some cases. Thus, users should have a control of their availability to secure their privacy. In our system, Cell phones are proactively and constantly looking for Beacons, and actively seek requests to fulfill. This ensures that the user is in complete control and can opt-out if necessary. If a user does not wish to participate in this exercise, the user just needs to turn off the functionality necessary for this system. Moreover, the SNS can limit the number of requests sent to a single user in a specified period (e.g. one per day) so that the user will not be overwhelmed with requests.

(Problem 4) Users may agree to buy an item and then decide not to or may decide to purchase an item only if no one else is available.

Users may want to decide to agree to complete a to-do with checking other users' status. For instance, a user may agree to purchase a book even if someone else has already agreed to purchase it because he/she can expect to complete it earlier. Therefore, simple two-phase commit is not enough; it may lose opportunities for efficient shopping coordination. Thus, we believe that additional context information should be provided for a user to make a decision for purchasing. We develop a user interface which allows multiple users to know the context of each other and negotiate who will purchase the drink. We will explain this in Distributed Two-phase Commit section later.

SYSTEM ARCHITECTURE

As mentioned earlier, our system consists of three major components; Cell phones, Beacons and SNS. In this section, we explain the features of our system. One feature is to allow the friends of a user to complete a to-do based on their context. To realize this, a user can post and share a to-do with his/her friends in SNS. A distributed two-phase commit allows the collocated users to coordinate the purchase collaboratively so that the system can maximize the opportunity of the completion of a to-do and can minimize the risk of "over-commitment" to the same to-do.

Completion of a To-do

We present a simple walkthrough to illustrate how our proposed system works and what messages are passed among the components in the system. In this scenario, Mary wants a latté and John is going to a coffee store soon.

Mary logs into SNS website and posts a to-do for a latté (Figure 3(1)). The posted to-do is shared with Mary's friend.

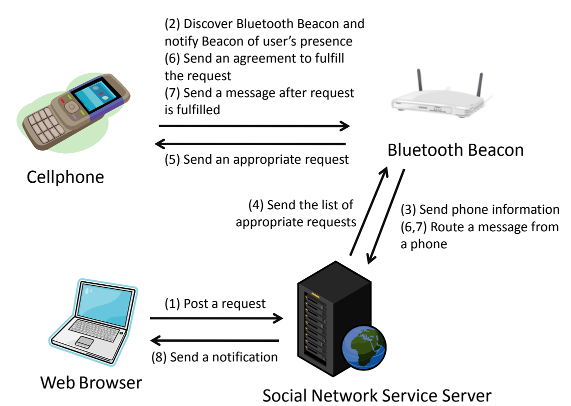


Figure 3. The flow of the completion of a shared to-do item on the Social Network Server.

John, one of Mary's friends, walks into a coffee store, and his Bluetooth-enabled phone registers his presence with Beacon that is present at that location (Figure 3(2)). This Beacon passes John's phone ID, along with its own location information to SNS (Figure 3(3)), which replies with a list of open requests that are applicable to that beacon's location and Mary's friendship (Figure 3(4)). The beacon examines the list of requests, and finds that John's phone is appropriate to notify about Mary's request. The beacon thus contacts John's phone and tells him about this drink that Mary wants (Figure 3(5)).

John decides to purchase the drink for Mary and confirms this through the cell phone's UI. A message is send to the SNS via the beacon and Mary's request is locked so that no other people will buy Mary a latté (Figure 3(6)). John approaches the counter, orders, pays and receives the drink. John then taps the appropriate button on his phone to confirm with the system that he has the drink in his possession. Beacon is notified, which informs the SNS, which updates the status of the drink order (Figure 3(7)). Mary is notified accordingly (Figure 3(8)). John brings the drink to Mary and the necessary payment is given.

Building a Social Network and To-do Sharing

First, a user must register in SNS to share to-dos. A user is required to input personal information, such as the name and the phone number, and the MAC address of the phone. The MAC address is used for resolving the user's identity automatically when the user gets in proximity to Beacons. Therefore, a user does not need to input any information to notify his/her location. After the registration, users can connect to their friends as in other online social networks.

Every user can post what they want to purchase in SNS. Using a web-based form, a user describes what he/she wants and selects an appropriate place, such as a café or a grocery store (Figure 4). The posted to-do is automatically shared with his/her friends. The posted to-do can be edited anytime unless one of the friends has accepted to complete it.

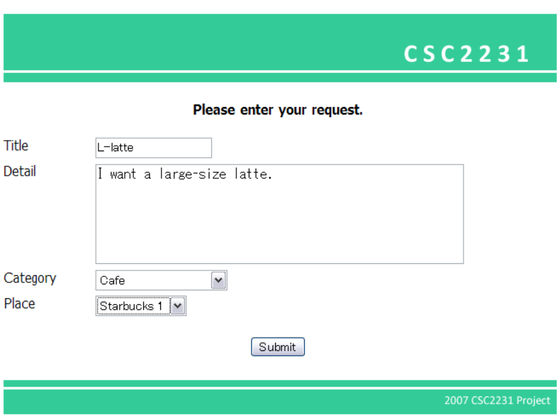


Figure 4. Web-based user interface for entering a shared to-do

Distributed Two-phase Commit

The traditional approach to solving the shared global variable problem is to use the two-phase commit. Had we used a two-phase commit, our system might work something like this:

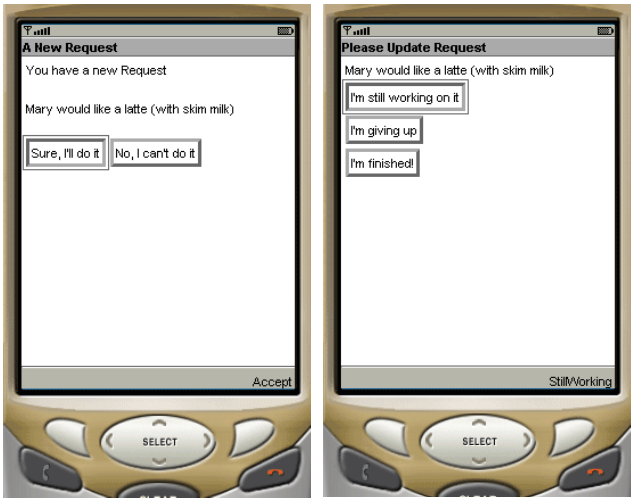
1. Users A, B, C receive drink purchase requests from users D. The first user who "accepts" the request wins the lock on the request, and is therefore allowed to service the request.
2. User B wins and is informed that he has won the lock, and hits "accept" to finalize his decision. User B goes and purchases the drink. If he chooses to back out at this point, the process returns to step 1.
3. User B completes the transaction, and updates the status on the social network server.

The classic two-phase commit protocol is blocking. While this is a sound solution to the problem from a systems perspective, we believe it fails to address some serious usability concerns. Once user B obtains the exclusive lock in step 2 above, the other users no longer have a chance to purchase the drink.

Thus, our system returns control of the locking mechanism to the users. In the example above, user B has accepted the task of purchasing D's drink. If any of D's other friends (users A and C) enter a location in which they too can perform this request, the system treats this arrival as an event and asks user B for an update on the task status. At this point, B has the option of backing out of the request or asking for more time to complete the request. If B gives up, the new user (A or C) is asked if they can perform the request.

User Interface

The user interface on Cell phone is very minimal so that interactions with the system are lightweight (Figure 5). It is our belief that a cumbersome user interface or numerous interactions with the application will result in users rejecting the system. There are two states in which the cell phone polls the user for an answer, when a request is first made and when a confirmation is sought by the beacon, such as in the case outlined in the previous section. While a user is fulfilling a request, a status screen shows the details of the request and includes a button that allows the user to update the request's status.



Request Screen **Confirmation Screen**

Figure 5. Cell phone UI for new requests (*left*), and updates (*right*)

CONCLUSIONS AND FUTURE WORK

In this paper, we presented a prototype system that leverages the opportunities that naturally arise in the course of the day to help one's friends out by purchasing items from their to-do lists. It is our hope that the widespread use of such a system can save its users an untold number of hours of traveling between shopping locations; traveling that is redundant when one's social network is leveraged.

This work is still in progress; there remain several things to be done as follows.

Evaluation

We would like to observe users as they interact with the distributed two-phase commit UI in a practical setting. What new opportunities have been enabled? We believe that additional contextual information (beyond ETC) is likely necessary and foresee additional work to look at the integration of other pieces of context for users.

Bluetooth Communication

We plan to modify the communication between beacon and phone to better address the power constraints on the cell phones. Currently, cell phones actively search for discoverable beacons, and we plan to switch these roles in our prototype so that the power-consuming searching is performed by the beacon and not the phone.

Improvements to Coordination Among Users

Users who are considering making the drink purchase report on the time they estimate it will take them to make a particular purchase. As the purchase process continues (as they wait in line), they can easily update their estimated time to completion (ETC) and all other users who received this request also receive this updated information. By allowing users to report this additional piece of context, we believe that users will be able to work collectively to reach an optimal solution.

Improvements to Reliability

Problems arise with our prototype when a user who has accepted a to-do either leaves the communication range of Beacon or experiences a failure of their cell phone. In this case, the system has no way of knowing whether the to-do was completed successfully. Because the cell phone application knows if there is an open request that was accepted by the user and also knows if it has left the range of Beacon, an improved cell phone application could use one of the other communication channels available to the phone to send an update to SNS. For example, after prompting the user for an update on the task status, a cell phone that is out of range of a beacon could send a simple SMS message with the task status to the SNS so that the system can either close the task or try assigning it to another member of the network.

Online Social Network Integration

Integration with Facebook in the future would allow this system to be more widely tested and refined as we get more user feedback.

ACKNOWLEDGMENTS

We would like to thank Khai N. Truong his comments on this project, and for supplying cell phone hardware for development purposes.

REFERENCES

1. dodgeball.com, Google Inc. http://www.dodgeball.com/
2. Kim, S. W., Kim, M. C., Park, S. H., Jin, Y. K., and Choi, W. S. (2004) Gate reminder: a design case of a smart reminder. In *Proceedings of DIS '04*, 81-90.
3. Ludford, P. J., Frankowski, D., Reily, K., Wilms, K., and Terveen, L. (2006) Because I Carry My Cell Phone Anyway: Functional Location-Based Reminder Applications. In *CHI Proceedings '06*, 889-898.
4. Meetup. Meetup Inc. http://www.meetup.com/
5. Multaharju, M., Koskinen, K., and Ikonen, J. (2004) Floating Note-A Location Based Messaging Application, In *Proceedings of WAWC '04*, 9-16.
6. Plaisant, C., Clamage, A., Hutchinson, H. B., Bederson, B. B., and Druin, A. (2006) Shared family calendars: Promoting symmetry and accessibility. *ACM Trans. Comput.-Hum. Interact.* 13:(3), 313-346.
7. Sohn, T., Li, K., Lee, G., Smith, I., Scott, J., and Griswold, W. (2005) Place-Its: A Study of Location-Based Reminders on Mobile Phones. In *Proceedings of Ubicomp '05*, 232-250.