MobiShop: Using Mobile Phones for Sharing Consumer Pricing Information

Shitiz Sehgal, Salil S. Kanhere, Chun Tung Chou
School of Computer Science and Engineering,
The University of New South Wales, Sydney, Australia,
Email: {sseh669, salilk, ctchou}@cse.unsw.edu.au

***Abstract*— Sensor networks provide tremendous potential for information collection, analysis and processing in a variety of application domains. The first-generation sensor networks were primarily a collection of stationary devices sensing ephemeral features of the environment around them. The rapid advancement in mobile technology combined with a surge in device penetration has fuelled the development of a new sensing paradigm i.e. *Participatory Sensing*. This paradigm is being increasingly used to design new and innovative applications, set to revolutionize present knowledge-sharing techniques. In this demonstration, we present a novel, people-centric application termed *MobiShop*, which facilitates sharing of product pricing information amongst consumers. Our system has been implemented on an off-the-shelf Nokia N95 mobile phone.**

I. Introduction

Price dispersion of retail goods is widely prevalent in our economy. Consumers who have access to the pricing information can benefit economically, as shown in studies such as [1] (up to 16% savings in purchasing electronic goods on-line). Several websites exist which enable consumers to compare prices of various retail goods. Examples include `Shopbot`[1] and `Pricescan`[2]. However, most of these websites only include pricing data from on-line stores, thus excluding most brick and mortar businesses, which may not have an on-line presence. *MobiShop* is a distributed computing system designed to collect, process and deliver product pricing information from street-side shops to potential buyers, on their mobile phones. In addition, it can also serve as an effective indirect advertising medium for retail shops. The system exemplifies a self-reliant, handheld mobiscope [2], in which, the subscribers are ultimately responsible for updating the information repository.

In recent years, several initiatives have used the concept of participatory sensing [3] to collect information about the urban surroundings, e.g, road traffic conditions [4] and cyclist experience [5]. However, MobiShop and our sister publication [6] are the first instance of a sensing system being used for sharing consumer pricing information. Our system is motivated by the success of applications such as Wikipedia, Youtube and BitTorrent, which are driven by altruistic participation of users.

II. System Design

Our system is implemented as a client-server program and has two principal modes of operation: (i) product price collection and (ii) user query. To contribute information, the user captures a digital image of the store receipt, which lists the products and corresponding prices using the mobile phone camera. MobiShop implements Optical Character Recognition (OCR) on the mobile device to extract the pricing information from the image. The products and prices are uploaded along with the GPS (Global Position System) coordinates of the user and the time of purchase to the central server. The server collates the user inputs and maintains an updated repository of the product prices at different stores. This database is interfaced to a GIS (Global Information System) street map populated with store locations. In the query mode, a MobiShop user can request for the prices of a particular product in her neighbourhood. The query sent to the server includes the GPS location of the user. The server replies back with a list of the stores featuring this product along with their prices in the vicinity of the user. Thus, MobiShop can be an effective aid to its subscribers in making informed decisions, while shopping. In the following we briefly provide details of the implementation of the client and server programs:

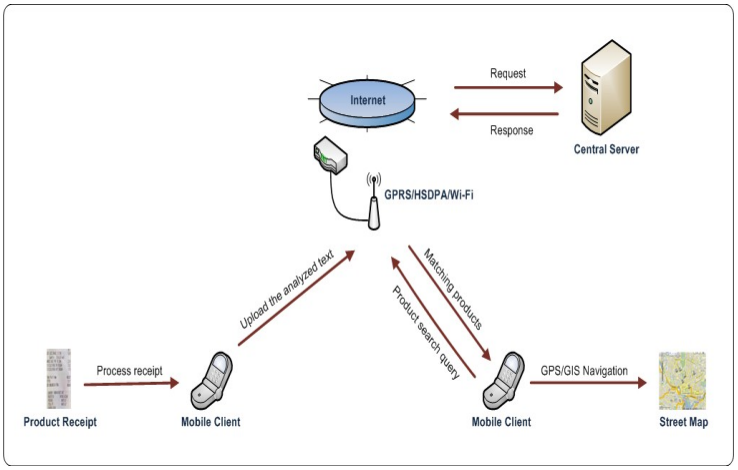

Fig. 1. System Architecture

Client: Fig. 2 depicts a logical representation of the various components that make up the client program. The MobiShop client software interface has been primarily implemented in Java ME to ensure portability across devices. However, Java ME does not currently provide any support for OCR, an important component of our system. Consequently we have

[1] `http://www.shopbot.com.au`
[2] `http://www.pricescan.com`

used the native Symbian OS 9.2 OCR engine to implement the requisite functionality. The image of the store receipt is passed as an input to the OCR program. The output consisting of the product names and prices is stored in a text file as illustrated in Fig. 3. We have tested the OCR program with images of 17 receipts from different stores. The OCR can accurately detect about 60% of words (the price of each item is considered as one word) on the receipt. In our future work, we intend to improve the accuracy of the OCR. Given the slightly high probability of errors, the user is given an option to edit the extracted text to fix some of the mistakes. This also allows the user to delete personal information such as credit card details, which may have been captured in the image. Our system leverages on the existing communication infrastructure for effecting the transfer of the product prices. Current mobile phones have ubiquitous Internet connectivity via the GSM/GPRS/3G/HSDPA cellular network and in some cases also via in-built 802.11 interfaces. The network communication unit of the client program establishes a TCP connection with the central server using any of the available underlying access technologies. The prices are uploaded to the server using this reliable channel. Since the images are processed on the mobile device, the client only needs to upload a few bits of data, thus minimising the upload costs. The GPS location of the user is obtained from the in-built GPS receiver (most new mobile phones are equipped with GPS receivers) and is uploaded with the text along with the current time.

The client also operates in query mode enabling the user to request for the latest pricing information for a particular product. A simple GUI is provided for user input. The user location is determined by querying the GPS receiver. The query is passed to the server daemon over a TCP connection. The server replies with a list of stores and the corresponding prices in the vicinity of the user. The client interfaces with an external GIS library, so that the store locations can be highlighted on a street map for navigation. We have implemented a prototype of the client program on a Nokia N95 8GB mobile phone.

Server: The central server is the main repository for storage and maintenance of the pricing information. The server program is written in Java and is executed as a daemon on an always-on workstation. We assume that the server has *a priori* knowledge of the store locations in the form of their GPS coordinates. This information could be readily obtained from a GIS system (for example TomTom already includes positions of several points of interest). In the future we also plan to have a web interface for registration of stores. When the server receives pricing information from the client, it first extracts the users GPS location and searches for the appropriate store. Note that, in our current implementation we assume that each location can have at most one unique store. In the future, we plan to include co-located stores and query the user to select the appropriate store. Alternately, the store name could also be extracted from the product receipt. The server then records the pricing information and replaces any older prices with the latest updates. When the server receives a query,

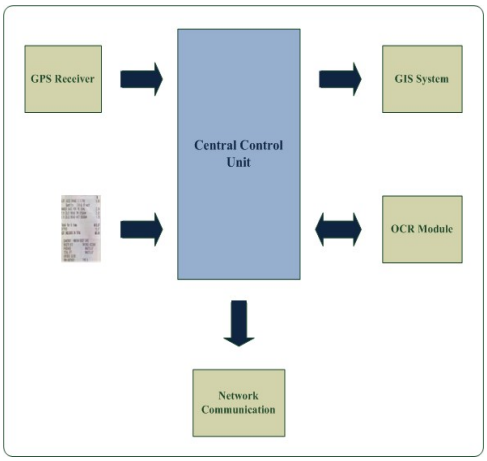

Fig. 2. Client Software

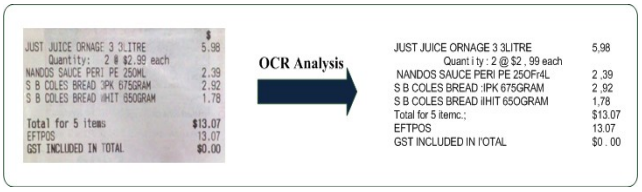

Fig. 3. OCR Output

it searches the vicinity of the user for stores which stock the requested product. A reply consisting of the store names and GPS locations is sent back to the client. In our current prototype, the server program is implemented using Tomcat and MySQL, running on a Windows/Linux platform. However, a real-world deployment would require Java EE compliance for scalability and efficiency.

ACKNOWLEDGEMENT

We would like to thank Rajib Rana, Yifei Dong and Assad Mehmood for helpful discussions. This work is supported by the Australian Research Council Discovery Grant DP0770523.

REFERENCES

[1] M. Maye, J. Morgan, P. Scholten, "The Value of Information in an Online Consumer Electronics Market", in *Journal of Public Policy and Marketing*, 2003, pp. 481-507.

[2] T. Abdelzaher, Y. Anokwa, P. Boda, J. Burke, D. Estrin, L. Guibas, A. Kansal, S. Madden, and J. Reich, "Mobiscopes for Human Spaces", in Pervasive Computing, vol. 6, no. 2, AprilJune 2007.

[3] A. Parker, S. Reddy, T. Schmid, K. Chang, G. Saurabh, M. Srivastava, M. Hansen, J. Burke, D. Estrin, M. Allman, and V. Paxson, "Network System Challenges in Selective Sharing and Verification for Personal Social and Urban Scale Sensing Applications" in *Proceedings of the 5th Workshop on Hot Topics in Networks (HotNets-V)*, November, 2006.

[4] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, "CarTel: A Distributed Mobile Sensor Computing System", in *Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems (SenSys2006)*, November 2006.

[5] S. Eisenman, E. Miluzzo, N. Lane, A. Peterson, G. Ahn and A. Campbell, "The Bikenet Mobile Sensing System for Cyclist Experience Mapping", in *Proceedings of the 5th ACM Conference on Embedded Networked Sensor Systems (SenSys2007)*, November 2007.

[6] Y. Dong, S. S. Kanhere, C. T. Chou and N. Bulusu, "Automatic Collection of Fuel Prices from a Network of Mobile Cameras", to appear in *Proceedings of the 4th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS 2008)*, June 2008.