Rapid Development of Civic Computing Services: Opportunities and Challenges

Shin'ichi Konomi[1], Kenta Shoji[2], and Wataru Ohno[2]

[1] Center for Spatial Information Science, The University of Tokyo,
Kashiwa, Chiba, 277-8568, Japan

[2] Department of Socio-Cultural Environmental Studies, The University of Tokyo,
Kashiwa, Chiba, 277-8563, Japan

{konomi,wataru_ohno,nebosukesan}@csis.u-tokyo.ac.jp

Abstract. Designing the right computing service for citizens can be extremely difficult without participatory and iterative service development processes. We discuss opportunities and challenges for quick, participatory service development by citizens, based on our experiences with two experimental context-aware services.

Keywords: civic computing, open-source hardware, data, collaboration.

1 Introduction

The ubiquity of computing and sensing technologies in everyday life is giving rise to a wide range of context-aware services for citizens, ranging from location-based social networking and navigation support to disaster information sharing and persuasive green-computing applications. Given the diversity of the target population and the complexity of the situations of use, the "one-size-fits-all" approach of providing a general service for everyone can limit the usefulness of the service significantly. To create useful context-aware civic computing services that suit different situations and needs, it is highly desirable to support citizen participation in the service development processes.

An important obstacle in enabling such participatory service development is the difficulty of handling context in civic computing systems. Although software architectures, such as Context Toolkit [1], can remedy some of the difficulty, context handling has other aspects besides software architectures, including hardware setup [2], data processing, and user experiences. In this context, we discuss a method to facilitate hardware setup by supporting exploration and contextualization of sensor data, and a strategy to facilitate data processing. The proposed method and strategy complements software architectures-based approach, thereby facilitating participation of people without the knowledge of programming, physical computing, or data processing/mining.

N. Streitz and C. Stephanidis (Eds.): DAPI/HCII 2013, LNCS 8028, pp. 309–315, 2013.

2 Developing Civic Computing Services

Civic computing provides digital services for various kinds of people who live in a city. Importantly, it can support citizens to cope with ill-defined social problems, and end-user development [3] based on citizen participation is a key approach in this context. In recent years, there is keen interest in such participatory approach as exemplified by the emergence of open-data hackathons and crowdsourcing projects.

To exploit open data as well as open source hardware components in a civic computing service, its development process can be understood a bit differently from traditional context-aware applications. As shown in Table 1, the development process can include Exploration (Step 2) as well as Setup (Step 3) to exploit hardware and data-source components, as well as the other steps that can be supported by appropriate software architectures (e.g., widget-based [1] or blackboard-based architectures).

Table 1. Steps for developing a civic computing service for sharing information about urban congestion

1. Specification	"Report congestion levels on trains using smart phones, sensors, and train-network data"
2. Exploration	Use sensors (CO_2, acceleration, etc.) to capture data on a train. Also, record congestion levels on the train manually. Create a model to transform sensor readings into a congestion level. Additionally, identify train-network data and explore ways to exploit them.
3. Setup	Connect a CO_2 sensor to a microcontroller and a wireless communication module, and integrate it with a physical object such as a bag.
4. Acquisition and delivery	Write software on a smart phone to obtain data from the CO_2 sensor, convert it to a congestion level, and upload it to a server along with the current location information.
5. Reception and action	Smart phone application receives data from the server and displays congestion levels on a map along with the train-network information.

3 Supporting the Use of Open-Source Hardware

This section introduces a prototype civic computing service for sharing congestion information, which we developed by using open-source hardware components. Our experience with this prototype service suggested the need to support exploration and contextualization of sensor data, which leads to a discussion on a general approach to supporting the use of open-source hardware.

3.1 A Congestion Sensing Service

Providing information about congestion in public spaces, including public transportation, can help citizens get around in a city without unnecessary stress. We therefore developed a prototype civic computing service that allows people to post congestion information using smart phones and/or open hardware-based sensors. As shown in Fig. 1, a CO_2 sensor and a battery are integrated with a bag, which forwards the data to a smart phone via a ZigBee-Bluetooth gateway. The smart phone estimates the congestion level based on the sensor reading and uploads it to a server along with the current location information.

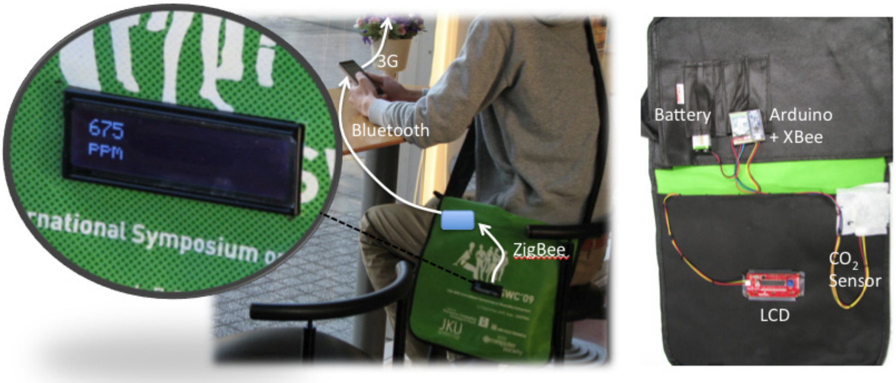

Fig. 1. Open hardware-based data capture environment for a congestion information service

3.2 Exploration of Sensor Data

To develop the congestion sensing service, we have explored the use of a CO_2 sensor, a pressure sensor, and an acceleration sensor. We expected that the CO_2 sensor could detect the increase of CO_2 concentration in a congested indoor space, and the other sensors could detect human posture (e.g., standing or sitting) that can be influenced by congestion. The result of this exploration suggested that we could use the CO_2 sensor only, whose measurements can be converted into estimated congestion levels relatively easily.

What we want to highlight here is the complicated work in this exploration process, which we cannot simply overlook in supporting the development of a civic computing service. As we are interested in providing the service for the users of a train network in Tokyo, we first obtained relevant information, such as the passenger capacity for different types of trains, from the railway company's website. We then developed simple data-logging devices using the sensors, and recorded various data in different congestion conditions, 11 times in total. We also counted the number of passengers in a train car for each data-recording sessions, and calculated actual congestion levels by dividing the number with the corresponding passenger capacity. Finally, we analyzed the data to discover the approximately linear relationship between the congestion levels and the CO_2 concentration.

3.3 Contextualization of Sensor Data

Past congestion information can become unreliable as people enter and leave the place of interest. For example, a congestion level reported by someone in a restaurant can become decreasingly reliable as people visit and leave the restaurant after the time of reporting. In this context, it can be useful to provide congestion information along with meta-information that indicates the reliability of the information. More generally, providing data along with relevant contextual information (e.g., reliability scores) can be a useful approach in various services.

In our prototype congestion information service, we devised an open hardware-based *"contextual sensor"* that can detect human flows in an entrance/exit area of a restaurant using an infrared sensor (see Fig. 2). It can be used to compute a reliability score of congestion information reported in the restaurant. Converting the sensor data into a reliability score requires a model that represents the correlation between the data and the score. This model was developed based on data recording sessions that resemble the ones we described in section 3.2.

Fig. 2. Prototype "contextual sensor"

3.4 Supporting Exploration and Contextualization

Clearly, it is a burdensome and time-consuming task to develop a civic computing service using open source hardware components, even though they are much easier to use than previous generations of ubiquitous sensing devices. That said, the following three approaches would minimize the burden to enable rapid prototyping, thereby facilitating participatory development by "expert amateurs" and other motivated citizens.

1. *Provision of a hardware toolkit*, which allow people to simply "plug in" and deploy different sensors without the knowledge of sensor network technologies. Such toolkit can be supported by web-based information sharing such as instructables.com.
2. *Support for crowdsourced data recording sessions*. Procedures for data recording sessions can be defined and shared so that a sufficient quantity of data can be collected quickly based on crowdsourcing.
3. *Support for sharing and reuse of data transformation models*. This can dramatically reduce the aforementioned burden since one could simply reuse an existing model.

4 Supporting the Use of Datasets

We now look into another prototype service, which exploits an existing dataset rather than sensor networks. The service recommends "nearby" restaurants based on a human-mobility dataset and the current location information. Deriving useful information for a civic computing service from a large-scale "general purpose" dataset can require multiple steps of costly data conversions, which could potentially discourage participatory development.

4.1 A Restaurant Recommendation Service

Existing location-based services that recommend restaurants simply based on Euclidean distances between the user and the restaurants do not consider other, more complex context of urban space, including patterns of transportation networks and human flows. We developed a prototype civic computing application that recommends restaurants in Tokyo, based on mobility patterns of crowds using the PFlow dataset [4]. Fig. 3 shows geographic areas that can be considered similar, based on people's mobility patterns in the dataset. As the service recommends restaurants based on this and other similar patterns, it can, for example, recommend a distant restaurant if it is located in a similar area that people at current location likely visit as well.

4.2 Data Processing

Deriving useful information for a civic computing service from a large-scale dataset such as PFlow can require multiple steps of data conversion.

Firstly, we divide the Tokyo region into small (1km x 1km) rectangular areas and find people who visit each area. Secondly, we generate a network of the rectangular areas using a similarity measure that is based on people's co-presence in the areas. Thirdly, we obtain clusters of areas based on the network, and finally use the clusters to generate restaurant recommendations.

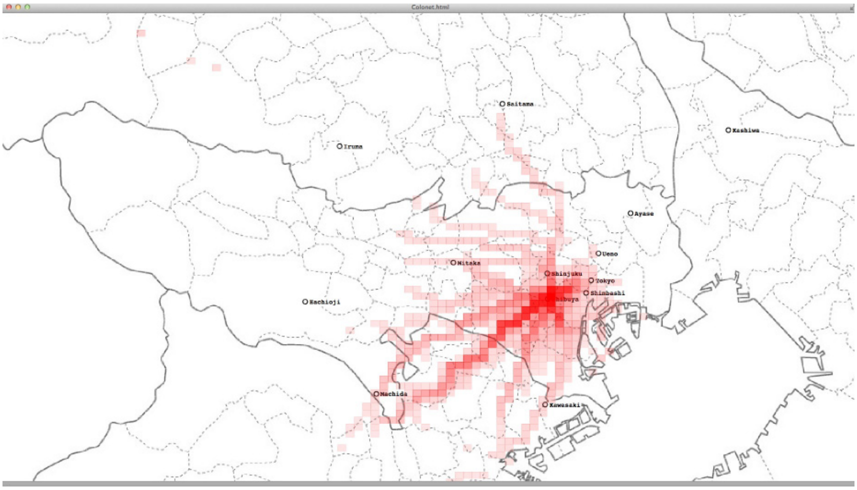

Fig. 3. Geographic areas that can be considered similar based on people's mobility patterns

4.3 Supporting Data Exploration and Processing

The data processing steps may appear somewhat simplistic. Nonetheless, it takes time (e.g., hours or days on a high-end desktop PC) to perform each step when the dataset is large. Moreover, each step requires the data processing person to determine several key parameters, including the size and shape of the rectangular areas, a threshold value for area-similarity judgment, and clustering parameters. The lengthy computation time inhibits casual modification and exploration of these parameters.

Here we suggest a couple strategies to cope with this issue:

4. *Publishing and sharing intermediate data conversion results.* It would be helpful to represent data conversion processes explicitly using a user-friendly (e.g., graphical) representations.
5. *Support for data exploration.* A tool for exploring the effects of different parameters using a small subset of the data would be extremely helpful.

5 Discussion and Conclusion

Our experiences with the prototypes suggested that the increasingly open sociotechnical environments (e.g., open source hardware, open data, open source, open access, etc.) create new opportunities to empower citizens in the development and use of civic computing services. We believe that effective support environment can be built based on the proposed methods and strategies, in order to address relevant challenges.

Acknowledgments. This research was supported by the Environmental Information project under the Green Network of Excellence (GRENE) program of MEXT, Japan.

References

1. Dey, A.K., Abowd, G.D., Salber, D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. Human-Computer Interaction 16(2-4), 97–166 (2001)
2. Beckmann, C., Consolvo, S., LaMarca, A.: Some assembly required: Supporting end-user sensor installation in domestic ubiquitous computing environments. In: Mynatt, E.D., Siio, I. (eds.) UbiComp 2004. LNCS, vol. 3205, pp. 107–124. Springer, Heidelberg (2004)
3. Fischer, G.: End-User Development and Meta-design: Foundations for Cultures of Participation. In: Pipek, V., Rosson, M.B., de Ruyter, B., Wulf, V. (eds.) IS-EUD 2009. LNCS, vol. 5435, pp. 3–14. Springer, Heidelberg (2009)
4. Sekimoto, Y., Shibasaki, R., Kanasugi, H., Usui, T., Shimazaki, Y.: PFlow: Reconstructiong People Flow Recycling Large-Scale Social Survey Data. IEEE Pervasive Computing 10(4), 27–35 (2011)