Context-Aware Mobile Crowdsourcing

Andrei Tamilin, Iacopo Carreras
CREATE-NET
Via alla Cascata 56/D, I-38123 Povo, Trento
{andrei.tamilin,
iacopo.carreras}@create-net.org

Emmanuel Ssebaggala, Alfonse Opira, Nicola Conci
University of Trento
Via Sommarive 5, I-38123 Povo, Trento
{emmanuel.ssebaggala,alfonse.opira}@studenti.unitn.it,
conci@disi.unitn.it

ABSTRACT
Ubiquity of internet-connected media- and sensor-equipped portable devices has emerged a range of opportunities for direct involvement of citizens into public decision making, leading to a new participatory format of public administration functioning. Intersecting the power of the crowdsourcing problem-solving paradigm by directly relying on human intelligence, with instantaneity and situation-awareness of mobile technologies, one gets a context-aware crowdsourcing approach for problem-solving in the right circumstances with the right people. In this paper, we present a prototype implementation of a context-aware mobile crowdsourcing system that enables the deployment and execution of crowdsourcing campaigns with users carrying mobile devices.The system is designed to maximize conditions for user participation, while minimizing the usage of energy. The paper describes the system architecture, defines an optimized sampling algorithm, and outlines a preliminary experimentation study carried out.

Author Keywords
Mobile crowdsourcing, context-aware systems, localization, energy-efficiency

ACM Classification Keywords
C.3 Special-purpose and Application-based Systems: Real-time and embedded systems

General Terms
Algorithms, Experimentation

INTRODUCTION
Active involvement of citizens in public decision making, such as urban planning and quality assessment campaigns of public services, has recently become a popular format of public administration functioning. This has been motivated by participatory sensing [1] approaches to objectively record, analyze, and discover patterns that are important in people's lives. As a result, public administrators have, nowadays, various instruments and platforms to let the users spontaneously voice their opinions.

At the same time, limited effort has been devoted to providing tools for administration to *crowdsource* citizens, and to know directly from citizens (or indirectly from citizen-created information sources) their opinions, emotional tonalities regarding certain arguments, and problems, as well as to seamlessly involve citizens in decision making.

In this paper, we present *sensoRcivico*[1], an integrated real-time civic awareness and engagement platform. The platform, conceptually depicted in Fig. 1, provides a technological solution for lowering communication barriers between citizens and public administration by taking advantage of the fact that the number of citizens possessing internet-connected media- and sensor-equipped smartphones is on the rise [3].

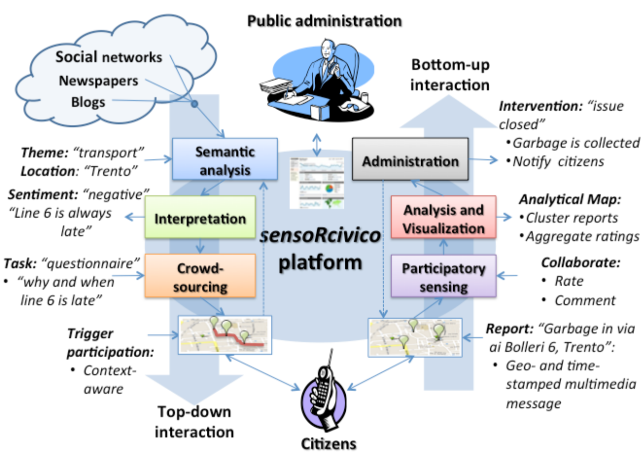


Figure 1. ***sensoRcivico*** **platform: bi-directional approach.**

The platform intersects a number of research and application domains, such as mobile technologies, participatory sensing and crowdsourcing in order to establish a technology-mediated *bi-modal interaction* channel between citizens and public administration. The bottom-up interaction modality enables citizens-to-administration communication, resulting in a participatory civic reporting system, in which citizens with mobile phones can submit thematic multimedia reports

[1] http://www.sensorcivico.it

UbiComp '12, Sep 5-Sep 8, 2012, Pittsburgh, USA.

on civic issues observed in the neighbourhood to the administration. Conversely, the top-down interaction modality enables administration-to-citizens communication, resulting in a mobile civic crowdsensing system [2], in which the administration can launch surveys or, more generally, tasks directly enquiring citizens.

In this work, we focus on the latter interaction by intersecting the power of crowdsourcing with instantaneity and situation-awareness of mobile technologies. The resulting system is able to deliver the right tasks to the right people in the right circumstances. The notion of context-awareness in the delivery and execution of tasks characterizes our approach and allows to (i) maximize conditions for user participation by presenting only tasks relevant to the user, with minimal user intervention (ii) minimize the consumption of resources of mobile devices, specifically, the battery, thus preserving their normal operation. In this paper we present the proposed system, together with the framework proposed for modelling context-aware mobile crowdsourcing systems. We introduce the algorithmic solution implemented to dynamically adapt crowdsourcing to user context. Finally, we validate it in a small field study, showcasing the potential of the proposed approach.

CONTEXT-AWARE MOBILE CROWDSOURCING SYSTEM

Our reference scenario targets a user carrying a smartphone with *sensoRcivico* crowdsourcing mobile application installed. The application runs in the background and periodically downloads from a server a list of tasks to be performed. The user is notified during his daily activities whenever one of the downloaded tasks matches the user's current context.

Depending on the application scenario, the context in which a given task should be presented to a user can be defined along multiple dimensions, such as geographical (e.g., within a circular area, along a street), temporal (e.g., in given dates, during given hours), user profile (e.g., age, gender), user activity (e.g., movement speed, no active calls), etc. The types of actions associated with a task can also vary with different application scenarios. An action can be a request for multimedia (e.g., an inquiry to take a photo, record a video or sound), participation in a questionnaire (e.g., answer a question, express a free-text opinion), etc.

Problem Formulation

A *mobile crowdsourcing task* t is a a tuple $t = \langle c^t, a^t \rangle$, where c^t is a *task context* scoping its applicability, and a^t is an *action* the task consists of.

For the sake of clarity, hereafter we consider the simplest structure of a task context defined by a geographical dimension, represented by a circular area, and temporal dimension, represented by a time interval:

$$c^t = \langle lat^t, lon^t, rad^t, [start^t, end^t] \rangle \tag{1}$$

where lat^t, lon^t are latitude and longitude coordinates of the circle center, rad^t is its radius, and $start^t, end^t$ are start and end timestamps.

The crowdsourcing system typically consist of multiple tasks and will be represented by a *tasks list* $\mathbf{t} = \{t_j\}$ of n tasks t_j, where $t_j = \langle c_j^t, a_j^t \rangle$, $0 \leq j \leq n$.

A user context, is an essential element of the mobile crowdsourcing system, determining if the user is in conditions relevant to (some) tasks. Similarly to tasks, a *user context* can be structured along multiple dimensions, which are practically determined by the sensing capacities of the user device. Hereafter, we assume the simplest structure of a user context c^u defined by a geographical dimension, represented by the current estimate of the user's location, and a temporal dimension, represented by the current time:

$$c^u = \langle lat^u, lon^u, acc^u, ts^u \rangle \tag{2}$$

where lat^u, lon^u are latitude and longitude coordinates of the user location, acc^u is the accuracy, with which the location has been obtained, and ts^u is the timestamp.

The list of chronologically ordered user context instances is a *user context history* $\mathbf{c}^u = \{c_i^u\}_{i \geq 0}$.

We define the *distance* $d^{u,t}$ between a user context c^u and a task context c^t as a function:

$$d^{u,t} = f_{dist}(c^u, c^t) \tag{3}$$

which depends on the dimensions constituting the contexts. In the case of contexts defined by (1), (2) this function can be computed as follows:

$$f_{dist} = \begin{cases} hvrs(lat^u, lon^u, lat^t, lon^t) & \text{if } ts^u \in [start^t, end^t] \\ \infty & \text{otherwise} \end{cases}$$

where $hvrs$ is the haversine formula for calculating the spatial distance between two latitude-longitude points.

Given a user with the context c^u, the task t is said to be *detected* by the user if $f_{dist}(c^u, c^t) \leq \lambda$, where λ is a distance threshold depending on a specific application scenario.

A user context sampling is the process of obtaining a user context by the mobile device. This process can be controlled by the following 2 parameters:

- sampling accuracy σ: this parameter adjusts the required distance accuracy for the context sampling. The higher its value, the more precise sampled context is.This parameter is particularly relevant for estimation of user location, because application of different localization sensors gives different estimation accuracy (e.g., GPS vs. cell towers);
- sampling rate ν: this parameter dynamically adjusts the time between any two consecutive context samplings. The higher its value, the more frequent the context is sampled.

A *user context sampling* is a function that, given a user context history and a list of tasks, estimates appropriate conditions for the next user context sampling:

$$\langle \sigma, \nu \rangle = f_{sampling}(\mathbf{c}^u, \mathbf{t}) \tag{4}$$

Given a task list $\mathbf{t}$, the goal of the mobile crowdsourcing system is to define the function $f_{sampling}$ that maximizes the number of crowdsourcing tasks properly detected and presented to the user, and minimizes the consumption of resources of the user's mobile device.

SYSTEM DESIGN AND IMPLEMENTATION

Implementation of the system essentially consists in defining the user context sampling, i.e., the function $f_{sampling}(\mathbf{c}^u, \mathbf{t})$. A key challenge to be addressed in implementing the sampling function lies in the optimization of the battery consumption in order to preserve the normal operation of the mobile device. In fact, when obtaining user location, battery consumption can be trade off for accuracy of the location estimation, depending on the type of sensor applied. In the present work, we consider the two most common smartphone localization methods: GPS-based, using the built-in Global Positioning System antenna, and cellular networks-based, using the relative position of the phone with respect to cellular towers visible to it. The typical GPS antenna of a modern phone gives an accurate location estimation, with an error value in tens of meters (or a couple of dozen meters in the worst case). However, GPS is also known to be energy intensive which implies significant battery drains [5, 6]. Conversely, the Cellular Networks method estimates the location with a high error of hundreds meters (or in worst case couple of kilometers), but with negligible battery consumption [4]. Intuitively, the idea behind the battery-efficient user context sampling consists in minimizing the use of the GPS antenna by maximizing the utilization of the cellular network localization method, when the corresponding localization error does not affect the quality of task detection.

The algorithm

The adaptive user context sampling algorithm is depicted in Alg. 1 and is graphically illustrated in Fig. 2.

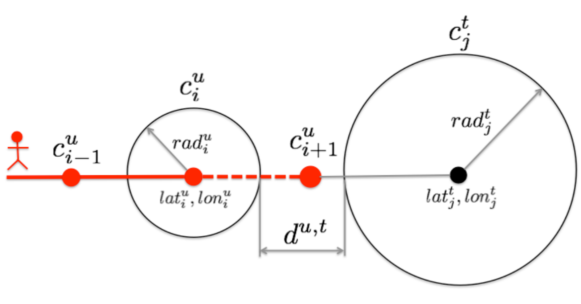


Figure 2. Adaptive user context sampling illustration.

The idea is to utilize the network whenever possible and switch to GPS and increase the sampling rate whenever the uncertainty in the user location overlaps with the spatial validity of the closest task. In other words, when $d^{u,t}$ becomes less than user location uncertainty or less than zero (Fig. 2) the mobile application localization switches to GPS, instead of the network. In such an approach network localization is utilized as a rough probe for checking if there are tasks in range, while the precise probing of near tasks is performed using GPS. In order to predict the next sampling time (ts_{i+1}), the algorithm computes the average speed of approach to the closest task using the context history and sets the value in anticipation of the overlap between user context and task context (this is regulated by the value μ). Tasks with negative $d^{u,t}$ are also marked as active so that the user is notified about possibility to execute them.

Algorithm 1 User context sampling function $f_{sampling}$

function NEXTCONTEXTSAMPLING($\mathbf{c}^u, \mathbf{t}$)
    $c_i^u \leftarrow getCurrentUserContext(\mathbf{c}^u)$;
    $c_{i-1}^u \leftarrow getPreceedingUserContext(\mathbf{c}^u)$;
    $c_j^t \leftarrow getClosestTaskContext(c_i^u, \mathbf{t})$;
    // Average approach speed wrt task t_j
    $\bar{v} \leftarrow \frac{|f_{dist}(c_i^u, c_j^t) - f_{dist}(c_{i-1}^u, c_j^t)|}{ts_i - ts_{i-1}}$;
    $d^{u,t} \leftarrow f_{dist}(c_i^u, c_j^t) - rad_i^u - rad_j^t$;
    if ($d^{u,t} \leq \lambda * rad_i^u$) **then**
        $\sigma_{i+1} \leftarrow$ 'GPS'; // Use GPS next
    else
        $\sigma_{i+1} \leftarrow$ 'NETWORK'; // Use Network next
    end if
    $ts_{i+1} \leftarrow \mu * \frac{d^{u,t}}{\bar{v}}$; // Next xampling time
    return $\langle \sigma_{i+1}, ts_{i+1} \rangle$;
end function

Prototype Implementation

The *sensoRcivico* context-aware mobile crowdsensing system has been fully implemented into a working prototype consisting of a server-side web application and mobile application for smartphones.

The server-side part has been implemented as a web portal providing an interface to public administrators to create tasks and monitor the process of their execution. For each task, the public administrator can configure: (i) its context, which in the current prototype implementation consists in the spatio/temporal region in which the the task should be triggered to users; (ii) the action requested to be performed by users. In particular, administrators could assign to the task a questionnaire, multimedia reports, or a combination of both. The server component receives user responses for each task, computes response statistics, and draws visual analytics for public administrators. Fig. 3 depicts the user interface of the server-side component.

In addition to the user interface, a RESTful API has been implemented on the server side to communicate with mobile clients to deliver tasks and receive task responses. A specific ontology has been developed in order to define a common language between the server and mobile clients, and to properly represent and interpret tasks and tasks responses'.

The mobile application has been implemented for Android-based smartphones using Titanium Appcelerator[2], a framework for mobile application development. The application downloads tasks from the server, schedules energy-efficient acquisition of the phone's context, and notifies the user about relevant tasks around, inviting him/her to participate. The context acquisition component uses GPS-based and Cellular Network-based localization capabilities of Android smart-

[2] http://www.appcelerator.com

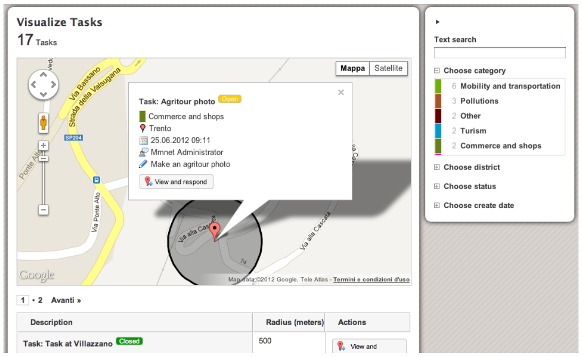


Figure 3. ***sensoRcivico*** **server-side user interface.**

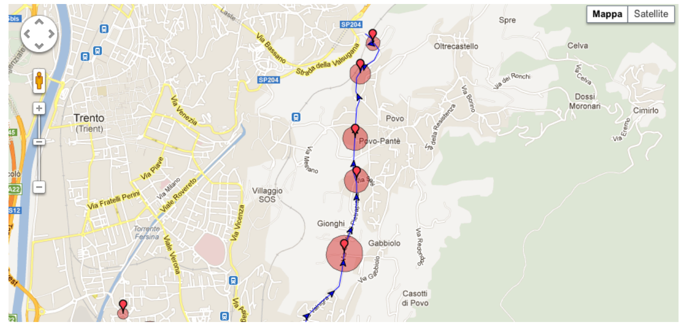
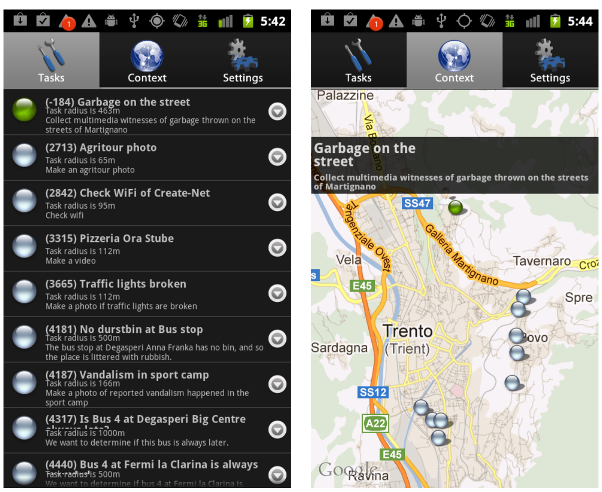


Figure 5. Experimental setting.

phones. The implemented multimedia capture component accesses the microphone and camera of the smartphone allowing the execution of multimedia actions assigned to the tasks, e.g., taking a photo, recording a video or a sound clip. Responses to tasks and acquired multimedia content are uploaded to the server for elaboration. Fig. 4 shows the user interface of the implemented mobile application.



(a) Tasks list. (b) Map.

Figure 4. ***sensoRcivico*** **mobile application user interface.**

PRELIMINARY EXPERIMENTS

In order to assess the feasibility of the implemented system and investigate the behavior of the implemented functions a small pilot was carried out. It consisted of a user moving along a predefined route by car, carrying a mobile device with the *sensoRcivico* application installed. Various tasks were pre-configured in the system along the route, each characterized by a given location and radius. This is illustrated in Fig. 5, where the blue line over the map corresponds to the route followed by the user, and the red circles characterize the task location and geographical validity.

Preliminary results confirm the ability of the system to properly detect tasks based on context and minimize the use of the battery-expensive GPS sampling. However, further refinement of the algorithm needs to be done in order to account for specific situations such as the time needed for acquiring the position of GPS, the non uniformity of speed and specific situations such as the absence of the GPS signal.

A large scale field trial is currently running in order to quantitatively evaluate each one of these aspects.

CONCLUSIONS

In this paper we presented *sensoRcivico*, a context-aware mobile crowdsourcing system which exploits user context in order to optimally present tasks to users, and preserve mobile device's resources. We presented the system design, together with the framework we used to model and evaluate the developed algorithmic solutions. Our preliminary evaluation encourages the proposed approach.

Current work is devoted to extending the dimensions utilized for characterizing the context, and implementing and evaluating large-scale experimentation involving a larger user base.

REFERENCES

1. J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory sensing. In *Proc. of WSW*, pages 117–134, 2006.
2. R. Ganti, F. Ye, and H. Lei. Mobile crowdsensing: current state and future challenges. *Communications Magazine, IEEE*, 49(11):32 –39, november 2011.
3. N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell. A survey of mobile phone sensing. *Comm. Mag.*, 48(9):140–150, Sept. 2010.
4. K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao. Energy-accuracy trade-off for continuous mobile device location. In *Proc. of MobiSys*, Jun. 2010.
5. J. Paek, J. Kim, and R. Govindan. Energy-efficient rate-adaptive gps-based positioning for smartphones. In *Proc. of MobiSys*, Jun. 2010.
6. C.-L. Wu, Y.-T. Huang, C.-L. Wu, H. hua Chu, P. Huang, and L.-J. Chen. An adaptive duty-cycle scheme for gps scheduling in mobile location sensing applications. In *Proc. of PhoneSense*, Nov. 2011.