Service-Mediated On-Road Situation-Awareness for Group Activity Safety

Amin B. Abkenar, Seng W. Loke and James Xi. Zheng
Deakin University
Burwood, Victoria, Australia
abakhsha@deakin.edu.au,seng.loke@deakin.edu
xi.zheng@deakin.edu.au

Arkady Zaslavsky
CSIRO, Data61
Melbourne, Victoria, Australia
arkady.zaslavsky@data61.csiro.au

ABSTRACT

Human activity recognition using embedded mobile and embedded sensors is becoming increasingly important. Scaling up from individuals to groups, that is, group activity recognition, has attracted significant attention recently. This paper proposes a model and specification language for group activities called *GroupSense-L*, and a novel architecture called *GARSAaaS (GARSA-as-a-Service)* to provide services for mobile Group Activity Recognition and Situation Analysis (GARSA) applications. We implemented and evaluated GARSAaaS which is an extension of a framework called *GroupSense* where sensor data, collected using smartphone sensors, smartwatch sensors and embedded sensors, are aggregated via a protocol for these different devices to share information, as required for GARSA. We illustrate our approach via a scenario for providing services for tour leaders aiding Vehicle-to-Human (V2H), Vehicle-to-Group (V2G) and Vehicle-to-Vehicle (V2V) interactions to increase the group safety. We demonstrate the feasibility of our model and expressiveness of our proposed model.

KEYWORDS

Context-Aware, Situation-Awareness, Pervasive Computing, Mobile Stream Data, Group Activity Recognition, Road Safety

ACM Reference format:
Amin B. Abkenar, Seng W. Loke and James Xi. Zheng and Arkady Zaslavsky. 2017. Service-Mediated On-Road Situation-Awareness for Group Activity Safety. In *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, Melbourne, VIC, Australia, November 7–10, 2017 (MobiQuitous 2017)*, 4 pages.
https://doi.org/10.1145/3144457.3145507

1 INTRODUCTION

Getting informed of occurred group activities, and situations within group activities, may benefit both a tracker (i.e., a person, vehicle, group or any consumer who is interested to receive information about groups) and a trackee (i.e., a person, vehicle or group being tracked who might also receive service/s from trackers). Collecting sensor data from a large number of individuals to aggregate, analyze and make sense of has been called Crowd Sensing (CS) [9]. But individuals may not have the tech-savviness and sufficient incentives to contribute or share their sensor data [4, 6]. Moreover, *Smart City* as one of the emerging domain of study intends to facilitate citizens' routine not only by automating services for them, but it can increase the quality of life by monitoring and providing real-time analysis of city entities [10]. A service-oriented approach to group activity recognition and situation analysis can be helpful to abstract away implementation details, where developers and users can view tracking and analyzing group activities as a service provided. This paper first proposes a high-level architecture of a framework called *GARSAaaS* (Group Activity Recognition and Situation Analysis-as-a-Service) and briefly presents its components. Then we introduce a new notion of *GA perspectives* which represents what a tracker wants to know about trackees, i.e., the perspective on the trackee activities, i.e., a group activity) of interest to a tracker. Next, for illustration and to demonstrate our approach, a prototype that implements two *situations* that can be tracked in a *'being in tour'* group activity will be presented. The contributions of the paper include:

- proposing GARSA-aa-Service design and its components for GARSA service provisioning, for consumers who are interested in being aware of aspects of group activities which benefits both trackees and trackers;
- a prototype and proof-of-concept implementation of the *GARSAaaS* architecture;
- a means to evaluate and test our prototype via building our own online synthetic sensor data generator (based on real data for multiple activities) for multiple users and cars' GPS coordinates.

2 MODELING AND CONCEPTS

We provide a model for group activities, based on a compositional view of group activities, that is, where a group activity is viewed as a composition and abstraction of the activities of the individuals in the group. Also, an architecture for the proposed framework and how they interact will be described.

2.1 System Overview

Figure 1 shows the high level architecture design of GARSA-aa-Service.

MobiQuitous 2017, November 7–10, 2017, Melbourne, VIC, Australia

ACM ISBN 978-1-4503-5368-7/17/11…$15.00
https://doi.org/10.1145/3144457.3145507

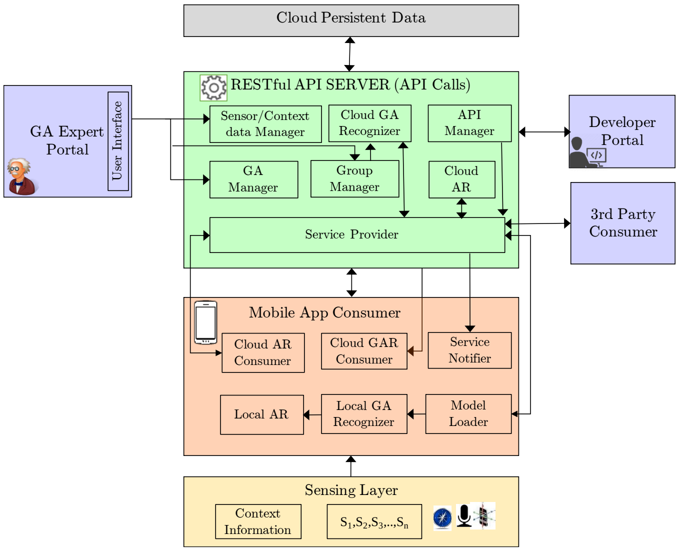

Figure 1: GARSA-aa-Service High level Architecture

Sensing Layer (Observer Side). The system receives sensor data from various kinds of sensors (e.g., mobile devices, embedded sensors in things, sensors in the environment).

Mobile App Consumer. This component of the system receives raw sensor data from the environment and also uses its own sensor data to recognize the performed activity via local AR (Activity Recognizer). There are two types of (activity/group activity) recognition process: local and cloud. In the local components, the recognition engine is located on the device whereas on the cloud component all the data is sent to the cloud and a cloud-hosted recognition engine executes the task.

RESTful API SERVER (API Calls). The key component of *GARSAaaS* is a RESTful API SERVER via which all the RESTful requests/responses to/from the platform are handled.

GA Expert Portal. This component is used by a domain expert to manage and add new group activity models, i.e. to define new GA expressions in the system. The GA Expert is a person who is able to model new GAs by exploiting *GroupSense-L* and can make sense of data to recognize a new activity. The expert is able to add new individuals and assign them to existing (or new) groups. Creating new situation rules is also part of this component's task. To build new activities, adding new sensors and context information may require defining sensor/context info specifications and linking them to a GA through the RESTful API Server.

Developer Portal. This component is mainly used by software developers to call *GroupSense* APIs in their products. GA developer may need to work with a GA expert to create new group activity models and include them in their applications.

3rd Party Consumer. In many cases, consumers want to use *GARSAaaS* to provide services to the target individuals and groups. They register in *GARSAaaS* and register their interest in groups, GAs and situations which are available.

2.2 Group Activity Modeling Language (*GroupSense-L*)

In our approach to group activity recognition and understanding, a Group Activity (GA) is first described in *GroupSense-L*, (EBNF below) which enables a precise description of the GA so that algorithms for its recognition and analysis can be programmed more easily.

$$GA = "(" \; Participants \; "," \; GA\_Name \; "," \; [\; Characteristics \;] \; "," \; Expression \; ")"$$

$$Expression = LTRA \; [\; CC \;] \; \{\; LTRA \; [\; CC \;] \;\} \; [\; RH \;] \; [\; RMMD \;] \; [\; RMMP \;]$$

In the above, *Participants* denotes GA participants, which can be people, vehicles or objects. *GA_Name* is the label of a GA or name of situation. In practice, each GA can be carried out in various ways and have different *GA characteristics*. Different people have different habits and it affects the way physical activities are carried out, called *GA characteristics*. *Logical Temporal Relationship Activity (LTRA)* denotes statements which describes performed activities by the individuals or group considering their logical/temporal relationships, e.g.:
"$((i_1, i_2, i_3 \bowtie walking)\ (b, 15s, 25s)\ ((i_1, i_2 \neg\ walking))$"
describes a situation where i_1,i_2,i_3 perform ("perform" denoted by $\bowtie$) walking activities between 15s and 25s $before$ (denoted by b) i_1 and i_2 are not ($\neg$) walking.
Context Condition (CC) produces conditions on context information, e.g.:
"$((i_1, location = "kitchen") \wedge ((i_2, i_3), distance > 10m))$"
simply defines location of i_1 in kitchen and ($\wedge$) distance betweeen i_2 and i_3 must be greater than 10m. Distance and location are context information in this example. Logical and Temporal relationships are applied on context conditions. The complete version of *GroupSense-L* is available[1].

2.3 A Model of Perspectives on Group Activities

Each GA consumer (tracker) might be interested to access information about a GA at different abstraction levels of the GA. Also, in order to protect the privacy of users (trackees) and provide services to trackees without invading his/her privacy, we introduce the notion of *perspectives*.

For example, in a tour group activity, *leader* is one role who should know about the trajectory information of tourists, while other trackers (such as museum or art gallery) only need to know if there are groups of tourists around their zones and their locations. Providing different information for different consumer's perspectives increases the flexibility of the system and helps protecting user's privacy. To implement the perspective model, a set of activity (GA/IA) attributes is defined for each activity to determine required resources to be recognized; selected attributes forms that perspective. Each activity attribute is associated with resources (sensor or

[1] https://goo.gl/ZnVdLK

context information). Formally, a perspective is a tuple of the form: $< Name, Activity\_Attribute\, list, Tracker\_Role >$

Similar to any other services/application there are terms and conditions which require users to share their data with the system. When a trackee joins certain groups in order to participate in certain GA (i.e., collaborative), s/he can also give permission to share his/her information relating to the GA; this could be in exchange for using a service.

3 *'BEING IN TOUR'* GROUP SAFETY RECOGNITION

This section illustrates the scenarios, used technologies, modelling and proof-of-concept experiments.

The aim of implementing this scenario is (1) to show the feasibility of *GroupSense-L* for representing and enabling detection of specifically represented situations, and (2) illustrating GARSA services both for the individuals/group and vehicles. GARSA services provided to the tour leader shown here are: monitoring spread of group (in order to avoid getting lost of members) ;and receiving an alert if any abnormal driving behaviors is occurred around the tour group. As mentioned in 2.2, we do not always aim to recognize a GA, and here, *being in tour* GA is already known but recognition of specific situations (which mostly triggers one service/action) within the group activity is the purpose of this experiment.

Used Technologies. Android SDK 24 was used for our implementation, and Android Wear SDK 21 used to exploit smartwatch sensors in our experiments. Google Cloud Endpoint is used as back-end server which is highly compatible with Google Android. The Apache commons Mathematics Java library [2] was used for feature extraction. We used the WEKA (Waikato Environment for Knowledge Analysis) data mining package[3] to build a classifier for individual activities; in particular J48 which is a Java implementation of the C4.5 algorithm.

Sensor Data Generator. In order to run our scenario and test on a larger scale while doing so in a realistic manner, a sensor data simulator called *GroupSense-Sim* was built to generate the required sensor data using seeding based on real data which was collected from a small group of users beforehand. The simulator is able to generate accelerometer data for given atomic activities (such as walking, running, standing, cycling and jogging) from two sources: smartphone and smartwatch. The simulator can generate random coordinates for individuals with a given starting point and a route path over a specified time period for a given activity. For example, by defining an expression and passing arguments: activity (for instance walking), start time, end time, route path and individual/group, the simulator generates GPS data. In a nutshell, we can run various scenarios by defining user/group behavior and *GroupSense-Sim* generates all the raw sensor/context data through separate Java threads. Each thread belongs to one individual and all the created threads for each sensor/context are handled by 'threads runner' which received all the threads and manages them.

[2] http://commons.apache.org/proper/commons-math

[3] http://www.cs.waikato.ac.nz/ml/weka

3.1 Situation 1. Tour Group Safety Monitoring - Group Spread

3.1.1 **Aim and Scenario.** Suppose a tour leader (the tracker) wants to prevent members from getting lost, to ascertain if the entire group (the trackees) is walking on track and to ensure individuals are not very spread out but keeping close together in the group. We can use the diameter of a group to provide an indication of how spread out the group is.

3.1.2 **Modeling.** A group being too spread out is represented in *GroupSense-L* as follows.

$$
\begin{aligned}
&Tour - group - spread - out = \\
&\quad ((g), BeingInTour, expression) \\
&expression = ((SP(g) > d)\ where: \\
&SP(g) = max\{distance(i,j) \mid (i,j) \in g'\}, \\
&\quad g' \subseteq g, g' = convex\_hull\_set(g)
\end{aligned}
$$

3.1.3 **Experimentation.** In order to measure the spread of the group, firstly, we compute the convex hull of the set of points of the group (i.e., the set of GPS coordinates of the group members) g'. To do that, the Graham Scan algorithm[4] is used to find the boundary point set. Then the maximum distance between any of pair of points in g' is calculated. The maximum distance represents the spread of the group. To evaluate our model, a new scenario was built by defining a group of 15 who are walking and getting close to and far from each other randomly for 30 minutes using our simulator. Every $60s$, the last $15s$ of users' location data is retrieved and our models executed and the detection algorithm ran over the users' coordinates. The value d (distance threshold expressed in the model) is set to $30m$. In the simulated scenario, the frequency of generating location data for users is set before building the scenario and there is no unexpected delay. Hence, the algorithm detects the situation with very high accuracy. However, in the actual devices, GPS data might have a few seconds lag time and also GPS inaccuracies may affect the accuracy of the spread recognition.

3.2 Situation 2. Tour Group Safety Monitoring - Preventing Accidents

3.2.1 **Aim and Scenario.** There is a chance when a tour group is visiting urban areas that the group is at risk of accidents with drunk drivers [2] or abnormal driving behaviors, e.g., exceeding the allowed speed or side-slipping [1].

$$
\begin{aligned}
&Tour - group - Abnormal - Driver = \\
&\quad ((g), BeingInTour, expression) \\
&expression = (1 \rfloor car \bowtie abnormal\_driving) \wedge \\
&\quad ((g, abnormal - car),\ distance < risk\_zone)
\end{aligned}
$$

3.2.2 **Experimentation.** In this experiment we assumed that the drivers (the trackees) uses some smartphone fixed in a phone cradle with a built in GPS, to report their locations to the service, and the tour leaders or tour group members (trackers), concerned about their own safety, subscribe to GARSAaaS to track the vehicles/drivers in the vicinity. However, obviously in practice, an

[4] https://courses.csail.mit.edu/6.006/spring11/rec/rec24.pdf

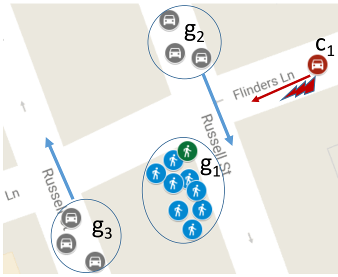

Figure 2: Example of Simulated Moving Vehicles and Tour Group

abnormal driver can refuse to use the proposed platform; but assuming all the cars will be soon equipped with motions sensors and GPS without the ability to disable them manually. The speed limit of the route is retrieved from **Google Road API** [5]. In this experiment, only speeds exceeding limits are taken into account as abnormal driving behavior. A random number of cars between 2-10 cars simulated (see figure 2) in different directions with respect to the tour group and one direction contains abnormal driving. Every 5s GPS data is collected and based on vehicle velocity and direction of vehicle, a distance estimation is calculated. The $risk\_zone$ value initially is set to 450m radius, i.e., as soon as a detected abnormal car reaches within 500m of the group, the tour leader receives an alert. After first run, we found out 450m is an appropriate value and gives sufficient time (approx 20s for a car which exceeds 20km/h) for the leader to take action. To increase safety margins, we can change the $risk\_zone$ to 650m in case there is any delay in messaging or the driver exceeds by more than 20km/h.

3.3 Situation 3. Traffic Clearance for Ambulance

Ambulance drivers must ensure the safety of other cars and people during high-speed driving. There is a high chance of clearing the traffic more effectively, if other drivers get notified of ambulance earlier e.g., 15s - 20s. We aim to implement this in our future work.

Group car crash also can be detected via the proposed framework.

By registering for events about ambulances (the trackees), via GARSAaaS, cars (the trackers) can get early warnings about the situation of emergency vehicles.

4 RELATED WORK

Group activity recognition in context-aware computing is a process of detecting or inferring of members' collective activity who are performing the same activity or collaborating and interacting to achieve a particular and more complex goal. A number of studies have been conducted in this area which can be categorized into two main categories: 1) vision-based [5], and 2) sensor-based [8]. Inherently, working with sensor data, especially mobile device sensor data, is very challenging. Accelerometer values are extremely sensitive to position. But the sensor-based approach has been used in many human activity recognition works in a wide-range of domains. In the Intelligent Transportation Systems (ITSs), safety applications have been designed to help decrease the severity of accidents (for V2V communication). However challenges such as delay in safety messages, channel congestion and the broadcast storm remained unsolved [3]. We propose a novel way of service mediated situation-awareness.

Gordon, et al. [7] defined MAR (Multi-User Activity Recognition) and differentiated it from GAR. They proposed a distributed architecture using coffee cups, each of which is equipped with sensors such as an accelerometer to detect whether individuals are drinking together or not. Our work explored the use of group activity recognition as a service, and outlined the *GARSAaaS* platform, illustrated via scenarios in the road safety domain.

5 CONCLUSION

We have proposed and implemented a prototype *GARSAaaS* platform, and evaluated and demonstrated *GARSAaaS* (Group Activity Recognition and Situation Analysis as a Service), a framework for continuous GAR using embedded sensors in mobile devices which provides recognition of GA or situation as services for clients. We note that *GARSAaaS* is capable of providing information from groups at different levels of abstraction for the consumers by using *GA perspectives*, providing a novel way for on-road situation-awareness for road safety for pedestrians and vehicles.

In the future work we will study the effect of GPS inaccuracy and delay on the accuracy of situation detection. Moreover, we will implement a smart watch alert system for all the nearby individuals, group and vehicles in the route when an emergency vehicle (e.g., ambulance, police) is heading to its destination in order to clear the traffic in a more effective manner.

[5] https://developers.google.com/maps/documentation/roads/speed-limits

REFERENCES

[1] Z. Chen, J. Yu, Y. Zhu, Y. Chen, and M. Li. 2015. D3: Abnormal driving behaviors detection and identification using smartphone sensors. In *2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. 524–532. https://doi.org/10.1109/SAHCN.2015.7338354

[2] J. Dai, J. Teng, X. Bai, Z. Shen, and D. Xuan. 2010. Mobile phone based drunk driving detection. In *2010 4th International Conference on Pervasive Computing Technologies for Healthcare*. 1–8. https://doi.org/10.4108/ICST.PERVASIVEHEALTH2010.8901

[3] Kashif Dar, Mohamed Bakhouya, Jaafar Gaber, Maxime Wack, and Pascal Lorenz. 2010. Wireless communication technologies for ITS applications [Topics in Automotive Networking]. *IEEE Communications Magazine* 48, 5 (2010), 156–162.

[4] M. Davari and H. Amintoosi. 2016. A survey on participant recruitment in crowdsensing systems. In *2016 6th International Conference on Computer and Knowledge Engineering (ICCKE)*. 286–291. https://doi.org/10.1109/ICCKE.2016.7802154

[5] Zhiwei Deng, Arash Vahdat, Hexiang Hu, and Greg Mori. 2016. Structure Inference Machines: Recurrent Neural Networks for Analyzing Relations in Group Activity Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[6] R. K. Ganti, F. Ye, and H. Lei. 2011. Mobile crowdsensing: current state and future challenges. *IEEE Communications Magazine* 49, 11 (November 2011), 32–39. https://doi.org/10.1109/MCOM.2011.6069707

[7] Dawud Gordon, Jan-Hendrik Hanne, Martin Berchtold, Ali Asghar Nazari Shirehjini, and Michael Beigl. 2013. Towards collaborative group activity recognition using mobile devices. *Mobile Networks and Applications* 18, 3 (2013), 326–340.

[8] Bin Guo, Zhiwen Yu, Liming Chen, Xingshe Zhou, and Xiaojuan Ma. 2016. MobiGroup: Enabling lifecycle support to social activity organization and suggestion with mobile crowd sensing. *IEEE Transactions on Human-Machine Systems* 46, 3 (2016), 390–402.

[9] L. G. Jaimes, I. J. Vergara-Laurens, and A. Raij. 2015. A Survey of Incentive Techniques for Mobile Crowd Sensing. *IEEE Internet of Things Journal* 2, 5 (Oct 2015), 370–380. https://doi.org/10.1109/JIOT.2015.2409151

[10] Albert Meijer and Manuel Pedro RodrÃŋguez BolÃŋvar. 2016. Governing the smart city: a review of the literature on smart urban governance. *International Review of Administrative Sciences* 82, 2 (2016), 392–408. https://doi.org/10.1177/0020852314564308 arXiv:http://dx.doi.org/10.1177/0020852314564308