STRAW Application for Collecting Context Data and Ecological Momentary Assessment

Junoš Lukan
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia
junos.lukan@ijs.si

Marko Katrašnik
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia
marko.katrasnik@gmail.com

Larissa Bolliger
Department of Public Health
Ghent University
Ghent, Belgium
larissa.bolliger@ugent.be

Els Clays
Department of Public Health
Ghent University
Ghent, Belgium
els.clays@ugent.be

Mitja Luštrek
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia
mitja.lustrek@ijs.si

ABSTRACT

To study stress at the workplace and relate it to user context and self-reports, we developed an application based on the AWARE framework, a mobile instrumentation toolkit. The application serves two purposes: of passively collecting data about user's environment and offering questionnaires as means of ecological momentary assessment. We implemented methods to import the questionnaires into the phone's database and trigger them at the right times. We also considered privacy implications of collecting such data and took additional measures to conceal the identity of our study's participants wherever we evaluated it was under the risk of exposure. Finally, we had to establish a server application to handle receiving and storage of collected data and implemented a rudimentary login process to additionally secure our servers.

KEYWORDS

context detection, application development, privacy, ecological momentary assessment

1 APPLICATION OVERVIEW

The best machine learning models for stress detection and affect recognition are multimodal [1, 17]. Combining data from different modalities is especially effective, such as using physiological, behavioural or contextual, and psychological (self-reported) data. Collecting such data in the real-world setting presents a challenge, however.

In the project called *Stress at work* (STRAW), the main objective is to analyse the relationship between psychosocial stress experiences in the workplace, work activities and events, and peripheral physiology. To facilitate integration of various data sources, an application was designed to run continuously and monitor their environment and specific phone-related events.

The application's purpose is two-fold. The primary mode of operation is silent and continuous: the user context (such as their phone use and location) is monitored without user intervention or interaction. The second mode of operation are prompts or questions for the user, where some information about the context and the participant's mental state is gathered by asking for it explicitly.

As a starting point for writing the STRAW application, we used AWARE, a mobile instrumentation toolkit which had the initial purpose of inferring users' context [5]. It enables logging of data as reported by the phone's operating system and a wide variety of hardware sensors. At several points, this toolkit was adapted to better suit our needs, and additional capabilities were added on top of it.

We also developed two modular functionalities of the application: Bluetooth integration with an Empatica E4 wristband [23] to enable simultaneous collection of physiological data and voice detection and speaker diarization capabilities [15]. We already reported on these developments elsewhere, whereas in this paper, we give an overview of the app's capabilities.

1.1 Data Types

An important aspect of the STRAW application are prompts, called EMAs. The users can be prompted to make a diary entry at a specific time which is called Experience Sampling Method [ESM; 3] or, more broadly (when data other than experience are noted), Ecological Momentary Assessment [EMA; 20]. Diary methods increase the reliability of collected self-reports as they are less prone to recall bias [14].

EMAs are the main mode of user interaction in the STRAW application. The content of specific questions is beyond the scope of this paper, but in general, the questions are based on existing psychological questionnaires measuring stressors, stress, and related responses. The implementation of EMAs is described in Section 2.

In addition to this, we selected a subset of data that might help us determine users' context. Below is a list of sensors that are used in the STRAW application together with the description of data they collect. Data availability from some of these sensors is dependent on phone's hardware and the version of operating system.

Information Society 2020, 5–9 October 2020, Ljubljana, Slovenia

- **Acceleration:** There are several sources (i.e. virtual sensors) of acceleration data in a smartphone. Accelerometers measure acceleration magnitude in various directions and report either linear acceleration (without gravity effects), gravity, or combined acceleration. This is used further in Google's activity recognition API [10].
- **Barometer:** Ambient air pressure.
- **Light:** Luminance of the ambient light captured by the light sensor.
- **Temperature:** Temperature of the phone's hardware sensor.
- **Bluetooth:** This sensor logs surrounding Bluetooth-enabled and visible devices, specifically their hashed MAC addresses, and received signal strength indicator (RSSI) in decibels.
- **Location:** Device's current location (latitude, longitude, and altitude, which are masked as described in Section 3) and its velocity (speed and bearing). This uses various methods, such as GPS and known Wi-Fis in vicinity resulting in different degrees of accuracy. Location category is also acquired with Foursquare API.
- **Network:** Network availability (e.g. none or aeroplane mode, Wi-Fi, Bluetooth, GPS, mobile) and traffic data (received and sent packets and bytes over either Wi-Fi or mobile data).
- **Proximity:** Uses the sensor by the device's display to detect nearby objects. It can either be a binary indicator of an object's presence or the distance to the object.
- **Timezone:** Device's current time zone.
- **Wi-Fi:** Logs of surrounding Wi-Fi access points, specifically their hashed MAC addresses, received signal strength indicator (RSSI) in decibels, security protocols, and band frequency. The information on the currently connected access point is also included.
- **Applications:** This includes the category of the application currently in use (i.e. running in the foreground) and data related to notifications that any application sends. Notification header text (but not content), the category of the application that triggered the notification and delivery modes (such as sound, vibration and LED light) are logged.
- **Battery:** Battery information, such as current battery percentage level, voltage, and temperature, and its health, as well as power-related events, such as charging and discharging times are monitored.
- **Communication:** Information about calls and messages sent or received by the user. This includes the call or message type (i.e. incoming, outgoing, or missed), length of the call session, and trace, a SHA-1 encrypted phone number that was contacted. The phone numbers themselves or the contents of messages and calls are not logged.
- **Processor:** Processor load in CPU ticks and the percentage of load dedicated to user and system processes or idle load.
- **Screen:** Screen status: turned on or off and locked or unlocked.
- **Voice activity:** A classifier, trained using Weka [7]. The features are calculated using openSMILE [4] and the output is an indicator of human voice activity [15].

The data described in the list above are collected automatically and continuously. The application is run as a foreground service, which means that the data collection continues even while the application is not actively used (i.e. it is minimized). Despite this, there exists software that is specific to the operating system version and phone manufacturer which tries to close applications for energy efficiency. We attempted to whitelist this application in the most common battery-saving software.

2 ECOLOGICAL MOMENTARY ASSESSMENT

As mentioned, one of the main functions of the STRAW application is to collect users' answers to questionnaires. AWARE already implements a 'sensor' for experience sampling method, which shows `DialogFragments` as the one in Figure 1, but it was too rudimentary for our study protocol. The main upgrades we had to make were the mechanism of triggering EMAs and management of the database of available questions (items) to include in the questionnaires.

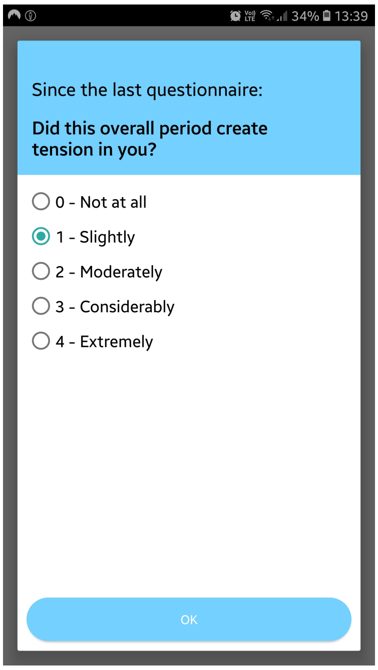

Figure 1: An example of an ecological momentary assessment prompt.

2.1 EMA Triggering

Originally, AWARE provides a couple of ways to trigger EMAs: at a specific time, by a certain context (i.e. taking into account values from other sensors) or on demand (manually). In our study, time is the most important trigger of EMAs, but we needed finer control.

The EMAs in our studies are divided into three types: a) morning EMAs with questions about sleep quality, b) work-hour EMAs with questions about momentary affect, job characteristics, work activities, and similar, and c) evening EMAs with questions about the whole workday and after-work activities. The first EMA is triggered in the first hour after the start of the workday as set by the user. The rest of the EMAs during work hours trigger approximately every 90 minutes, but not closer than 30 min apart. The time is dependent on the last answered EMA rather than set in advance, and additional reminders are scheduled in the case of user inactivity. The final EMA of the day is triggered in the evening at a time set by the user.

Each of these types of EMA is implemented as a separate `IntentService` [11] and handled by a `JobScheduler` [18]. This enabled us to enforce the requirements outlined above such as setting the minimum latency with which the job can start and making use of periodic jobs.

2.2 Question Database

In the original AWARE implementation, questions are queued into a questionnaire directly in the code of the application by using their custom `ESMFactory` class. For our study, we use a pool of more than 200 questions per language from which a subset is sampled for every EMA. We therefore needed a more systematic way of storing them within the application.

To ease the insertion of individual items, we prepared a spreadsheet template which is meant to be human-readable and filled out manually. Individual items from this spreadsheet are later converted into JavaScript Object Notation (JSON) and stored in an SQLite database [13] in phone's internal storage. This implementation enabled us to adapt the content of EMAs without touching the source code of the application. It also simplified the final selection of questions, such as selecting one language (English, Dutch, or Slovenian) and grammatical gender.

3 PRIVACY ENHANCEMENTS

The data collected by the STRAW application have different degrees of risk to the users' privacy. Their privacy would be threatened if an outsider gained unauthorized access to the data. These possible external threats are considered in Section 5.

Even when the data are safely communicated and stored, however, an involuntary exposure of users' identity might still be possible. Assuming the data are well protected from unauthorized external access, these risks will in turn be treated as internal in this section.

Some of the data collected by the STRAW application are personal data, so even when storing them securely and after pseudonymization, some risk of a privacy breach remains. Since AWARE is widely used in scientific studies it already implements some privacy enhancing mechanisms. We performed a thorough application vulnerability analysis and identified several further threats to privacy that we wished to address. While the data are safely communicated and stored, an involuntary exposure of users' identity might still be possible. The types of data that deserve special attention are applications, communication, location, and voice activity.

As mentioned in Section 1, the notifications that other applications send are monitored in the STRAW application. The content of the notification, such as that of an instant messaging application or calendar notification, is never actually stored. We deemed even the application names to be sensitive, so we chose to only save application categories. This process is further described in Section 4.

The content of calls or messages is never logged, but the phone numbers tied to them can be. Since we wanted to keep track of recurring contact with the same person, but not reveal their real phone number, we decided to encrypt them using the SHA-1 algorithm. While it would be possible to decrypt a phone number by a brute-force attack, the AWARE implementation offers the option of adding a salt. Thus by using the username (further described in Section 5) as a salt, the phone numbers are sufficiently protected from inadvertent disclosure risk, while the hashed value is retained even across different application installations. The MAC addresses of detected WiFi and Bluetooth devices are hashed in the same way.

The location data in their raw form are highly revealing of a user's identity [2]. Instead of storing the actual geographic coordinates provided by this sensor, the Foursquare Places API [6] is used to extract the category (venue) of a location. This API enables saving general categories such as 'bookstore' or 'gas station' near the user's location. But since we wanted to keep the option to analyse users' movements, we also implemented a transformation of coordinates. We converted longitude and latitude into spherical coordinates, applied a stochastic rotation (but constant within a specific user) and converted these back to transformed longitude and latitude. This enabled us to keep the distances between the locations faithful to original data, but transformed to another place on Earth.

As described in our previous work [15], voice activity recognition is performed on the phone in its entirety. This means that raw audio recordings can be discarded immediately after processing and only the calculated features are saved to the database. Alternatively, only the final binary prediction of human voice presence can be retained, but this makes any post-hoc analysis (such as speaker diarization) impossible.

4 SERVER APPLICATION

For the purpose of storing the data on a server, a Python application was implemented in Flask [21], which accepts the data in a JSON format and saves it in a PostgreSQL [22] database. In addition to receiving the data and managing credentials (as described in Section 5), it also performs a couple of additional functions.

As mentioned in Section 3, instead of saving application names we only log their category as classified in Google Play Store. To reduce the number of queries, we implemented this as a part of the server application. As part of the upload process, the application name is received in plain text, but only retained until query returns its category. After that, the application name is hashed to enable comparisons with later records and the name in plain text is discarded. In this way, we could build a database of application name hashes and their corresponding categories on the server, while not keeping a record of what applications individual users use.

The server application also provides a simple UI for administrators, where some metadata about the data collection itself are shown in forms of tables and charts. We can access data on last upload, number of days of participation, and number of data points for each individual user. This enables us to detect any problems with data collection and troubleshoot them early.

5 CLIENT-SERVER COMMUNICATION AND LOGIN

The STRAW application and other sensing applications are not special in the degree they could be subject to external attacks [2]. An attacker might want to expose identity of a user or try to reveal their personal data such as location. There are three points of entry for an external attacker: local storage, transmission of data, and the servers.

While the data reside on the device they are saved locally in the phone's storage. According to Android's documentation, this database is exclusive to the STRAW application [9]:

> Other applications cannot access files stored within internal storage. This makes internal storage a good

place for application data that other applications shouldn't access.

Additionally, once the data are transmitted to the server, the local database is periodically deleted. This reduces the privacy risk of the database being exposed, while also decreasing the local storage requirements.

It is therefore the transmission of data where we had to secure the data. They are transmitted over encrypted HTTPS connection, which eliminates the risk of exposure during this part of communication. The data are received by an application server residing at Jožef Stefan Institute (JSI), with a dedicated port listening for incoming transmissions.

The application server communicates with another, database server, also residing at JSI. This second server can only be accessed from within the JSI local area network. The database itself is also protected with a password and the user accessing it via the application server does not have administrator privileges.

Since the STRAW application is a part of a wider study, it is disseminated to recruited participants only. In addition to the data from this application, other data are collected, such as responses to questionnaires in baseline screening and physiological data from wristbands. It was therefore necessary that the data can be linked back to an individual in order to join the data from various sources. We developed a login method to enable this.

Using OkHttp [19] client-side and Flask-HTTPAuth [12] server-side, we implemented basic access authentication and token authentication [16]. The login credentials are disseminated to registered participants in our study and are input upon the installation of the STRAW application. This serves multiple purposes: by requiring login, we only accept data from actual participants of our study, while we can also use the assigned username to pseudoanonymously link data from various sources.

6 CONCLUSION

The application used in the STRAW project serves a dual purpose: to collect users' answers to questionnaires and passively collect data about their environment and phone usage. While the application was tailored to requirements of our study, this paper outlined the main issues and possible solutions when developing an application for research purposes.

The AWARE framework provided a solid foundation and especially eased sensor data collection, there are additional challenges that researchers need to face when trying to use an application like this in a scientific study. The data gathered using this application will help us develop improved models of stress recognition [8], which will help us integrate physiological data with more detailed contextual data and more reliable self-reports.

ACKNOWLEDGMENTS

The authors acknowledge the STRAW project was financially supported by the Slovenian Research Agency (ARRS, project ID N2-0081) and by the Research Foundation – Flanders, Belgium (FWO, project no. G.0318.18N).

REFERENCES

[1] Ane Alberdi, Asier Aztiria and Adrian Basarab. 2016. Towards an automatic early stress recognition system for office environments based on multimodal measurements. A review. *Journal of Biomedical Informatics*, 59, (February 2016), 49–75. doi: 10.1016/j.jbi.2015.11.007.

[2] Delphine Christin. 2016. Privacy in mobile participatory sensing. Current trends and future challenges. *Journal of Systems and Software*, 116, 57–68. doi: 10.1016/j.jss.2015.03.067.

[3] Mihaly Csikszentmihalyi, Reed Larson and Suzanne Prescott. 1977. The ecology of adolescent activity and experience. *Journal of Youth and Adolescence*, 6, 3, (September 1977), 281–294. doi: 10.1007/bf02138940.

[4] Florian Eyben, Felix Weninger, Florian Gross and Björn Schuller. 2013. Recent developments in openSMILE, the Munich open-source multimedia feature extractor. In *Proceedings of the 21st ACM international conference on Multimedia - MM '13*. ACM Press. doi: 10.1145/2502081.2502224.

[5] Denzil Ferreira, Vassilis Kostakos and Anind K. Dey. 2015. AWARE: Mobile context instrumentation framework. *Frontiers in ICT*, 2, 6, 1–9. issn: 2297-198X. doi: 10.3389/fict.2015.00006. https://www.frontiersin.org/article/10.3389/fict.2015.00006.

[6] Foursquare. [n. d.] Places SDK. Venue search. Retrieved 26/08/2020 from https://developer.foursquare.com/docs/api-reference/venues/search/.

[7] Eibe Frank, Mark A. Hall and Ian H. Witten. 2016. *The WEKA workbench*. (4th edition). Morgan Kaufmann.

[8] Martin Gjoreski, Mitja Luštrek, Matjaž Gams and Hristijan Gjoreski. 2017. Monitoring stress with a wrist device using context. *Journal of Biomedical Informatics*, 73, 159–170. issn: 1532-0464. doi: 10.1016/j.jbi.2017.08.006.

[9] Google. [n. d.] Access app-specific files. Access from internal storage. Retrieved 26/08/2020 from https://developer.android.com/training/data-storage/app-specific.

[10] Google. [n. d.] Adapt your app by understanding what users are doing. Retrieved 26/08/2020 from https://developers.google.com/location-context/activity-recognition.

[11] Google. [n. d.] IntentService. Retrieved 26/08/2020 from https://developer.android.com/reference/android/app/IntentService.html.

[12] Miguel Grinberg. [n. d.] Flask-HTTPAuth. Retrieved 26/08/2020 from https://flask-httpauth.readthedocs.io/en/latest/.

[13] D. Richard Hipp, Dan Kennedy and Joe Mistachkin. 2019. SQLite. Computer software. (2019). https://sqlite.org/index.html.

[14] Gillian H. Ice and Gary D. James, editors. 2006. *Measuring emotional and behavioral response. General principles. Measuring Stress in Humans. A Practical Guide for the Field.* .Part II –Measuring stress responses. Cambridge University Press, Cambridge, UK, (December 2006). Chapter 3, 60–93. isbn: 978-0-521-84479-6.

[15] Marko Katrašnik, Junoš Lukan, Mitja Luštrek and Vitomir Štruc. 2019. Razvoj postopka diarizacije govorcev z algoritmi strojnega učenja. In *Proceedings of the 22nd International Multiconference INFORMATION SOCIETY – IS 2019*. Slovenian Conference on Artificial Intelligence. Mitja Luštrek, Matjaž Gams and Rok Piltaver, editors. Volume A, 57–60. https://is.ijs.si/archive/proceedings/2018/files/Zbornik%20-%20A.pdf.

[16] Chris Schmidt. 2001. Token based authentication. In *Accepted papers for FOAF-Galway*. 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web. https://www.w3.org/2001/sw/Europe/events/foaf-galway/papers/fp/token_based_authentication/.

[17] Philip Schmidt, Attila Reiss, Robert Duerichen and Kristof Van Laerhoven. Wearable affect and stress recognition: a review. (21st November 2018).

[18] Joanna Smith. 2016. Scheduling jobs like a pro with JobScheduler. https://medium.com/google-developers/scheduling-jobs-like-a-pro-with-jobscheduler-286ef8510129.

[19] Square, Inc. 2019. OkHttp. Computer software. (2019). https://square.github.io/okhttp/.

[20] Arthur A. Stone and Saul Shiffman. 1994. Ecological momentary assessment (EMA) in behavioral medicine. *Annals of Behavioral Medicine*, 16, 3, 199–202. DOI: 10.1093/abm/16.3.199.

[21] The Pallets team. 2010. Flask. Computer software. (2010). http://flask.pocoo.org/.

[22] The PostgreSQL Global Development Group. 2019. *PostgreSQL 11.3 Documentation*. Version 11.3.

[23] Marija Trajanoska, Marko Katrašnik, Junoš Lukan, Martin Gjoreski, Hristijan Gjoreski and Mitja Luštrek. 2018. Context-aware stress detection in the aware framework. In *Proceedings of the 21st International Multiconference INFORMATION SOCIETY – IS 2018*. Slovenian Conference on Artificial Intelligence. Mitja Luštrek, Rok Piltaver and Matjaž Gams, editors. Volume A, 25–28. https://is.ijs.si/archive/proceedings/2018/files/Zbornik%20-%20A.pdf.