Scheduling Real-Time Acquisition of Context Information

Haonan Zhao, Fausto Giunchiglia

University of Trento, DISI, via Sommarive, 9, Trento, Italy

Abstract

Mobile self-reports, as a human-machine interaction technique, have gained popularity for collecting participant-labeled data in naturalistic settings. The integration of context-awareness into mobile self-report systems offers the potential for establishing symbiotic interactions between humans and machines. However, the quality of context information obtained through self-reports is often not good due to missing or inaccurate responses resulting from the busy of daily life or ill-timed questions. From a researcher's perspective, designing data collection events with different frequencies can be challenging due to the lack of a general methodology. To address this challenge, we propose a novel scheduling method designed to facilitate meaningful human-machine interactions by providing researchers with convenient for setting questions. Our new language employs the iCalendar standard and integrates a recurrence methodology to collect annotation of context data from both questions and sensors that may influence response quality, elicit high-quality responses, and enhance the overall user experience.

Keywords

Human-Machine interaction, Context, Scheduling language, Question representation

1. Introduction

The ability to process context information enables humans to adapt their behavior to their surroundings [1], as such, the acquisition of context information is essential. Due to the complexity of context, it is difficult to get high-quality context data directly. The challenges have significant implications for a range of fields, including Human-Centered Artificial Intelligence, Psychology, and Pervasive Systems [2, 3]. The necessary contextual information can be obtained from a variety of objective sources, such as networks, devices, sensors, and user profiles (see, e.g., [4, 5]). However, these objective data can make it difficult to capture the subjective motivations of participants. With the rise of Experience Sampling Method (ESM) [6], smartphones offer an opportunity to collect both objective (sensors) and subjective (self-reports) data from a user. This data can help machines learn from humans and assist users in improving their quality of life. Considering the send and receive information ability of the pervasive smartphones, they allow to collect sensor data and questionnaires anytime and anywhere, e.g., [7, 8].

However, questions are often sent to users at inopportune moments, such as during meetings or while driving, which can disrupt their daily routines and result in the collection of low-quality responses. This may also be due to recall bias, where participants do not accurately remember previous events [9]. Many studies have aimed to improve the response rate of questions, i.e., the number of answered questions, for example, see, [10, 11, 12]. Furthermore, Berkel et al. [13] showed that longer completion times for questions result in less accurate responses. However, these studies do not account for the flexibility of when users can answer questions. In other words, researchers typically send questions to users on a fixed schedule, as it can be challenging to design different frequencies for these questions. To the best of our knowledge, no research group is currently addressing this issue.

Therefore, it is necessary and valuable to develop a method that minimizes interruptions to users and give the flexibility to researchers for collecting effective data. This can be achieved by considering two key factors: when to ask questions and how to ask them. While the first perspective has been addressed in [14]. This paper focuses on providing researchers with the flexibility to design the schedule for collecting context data, including both questions and sensors, allowing them to obtain the information they prefer. Our scheduling language, which is an enhanced version of the iCalendar standard, allows participants to check when they need to provide feedback through their calendars, enabling them to match their real-life schedules and improve their experience, and consequently, the quality of their responses. For scheduled events in the Calendar, we can set different recurrence rules, defining the repeat pattern for different types of data collection, namely questionnaires and sensor collection. We propose a language for modeling questions based on their type and recurrence rule, providing users with greater flexibility in answering and establishing meaningful Human-Machine Interaction.

MRC 2023 – The Fourteenth International Workshop Modelling and Representing Context. Held at ECAI 2023, 30.09.-5.10.2023, Kraków, Poland.

haonan.zhao@unitn.it (H. Zhao); Fausto.Giunchiglia@unitn.it (F. Giunchiglia)

0000-0003-0825-7188 (H. Zhao); 0000-0002-5903-6150 (F. Giunchiglia)

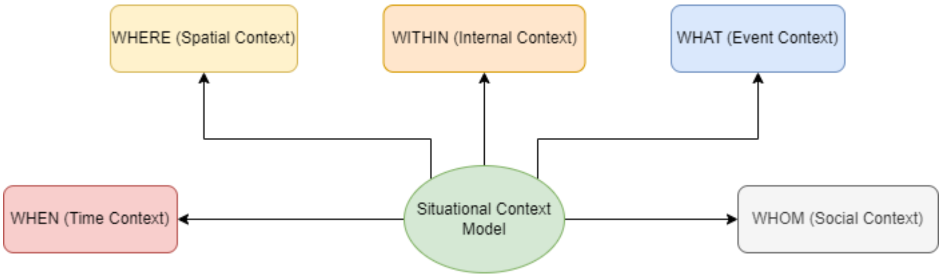

Figure 1: The context model.

The remainder of this paper is structured as follows. Section 2 presents the annotation context data collected through our scheduling methodology, mainly five key dimensions. Section 3 defines the BNFs of scheduling language, we divide into three parts, which used to schedule context collection. Section 4 reports on a case study that demonstrates the feasibility of our system, and how it can give the flexibility to the researchers. Finally, Section 5 concludes the paper.

2. Collecting Data about Context

The goal of asking questions or collecting sensor data is to understand the personal context annotation from a user. When discussing context, we consider both objective factors, such as time (measured through sensors), and subjective factors, such as moods (reported through self-reports), which together comprise what we refer to as the annotation context. However, other studies mainly collect context data through objective sources [4, 5]. To improve the accuracy of context data collection, it is necessary to address the inherent errors associated with smartphone sensor data. Therefore, we also propose obtaining answers and feedback from users directly to obtain annotated data, that is, we collect annotation of context data from two ways: *Questionnaires and Sensors.* Two key factors may impact the quality of the collected context data: the quality of the answers and the manner in which contextual information is elicited from users. In this paper, we focus on the latter and aim to establish meaningful human-machine interactions for the collection of context data. The annotated context used in this study is defined in [15, 16]. Specifically, our approach considers five key dimensions of context, as shown in Fig 1. Where: (1) *WHEN*, the temporal context, which defines the time at which our questions are presented to the user; (2) *WHERE*, the spatial context, which defines the location of the user when they respond to our questions; (3) *WHAT*, the event context, which defines the activities in which the user is engaged when they respond to our questions; (4) *WHOM*, the social context, which defines the individuals with whom the user is interacting when they respond to our questions; and (5) *WITHIN*, the internal context, which defines the emotional state of the user when they respond to our questions.

3. Scheduling for Context Data Collection

To improve the quality of personal context data, we propose using calendars to represent the activities that a user is expected to respond to context data collections. Different calendars can capture different aspects of a person's memory. For example, one calendar may record socio-demographic information while another may track event contexts. In this paper, we focus on using a single calendar to describe the real-time context of a user. We employ our own planning language data schema, rather than strictly following to the iCalendar/RFC5545 standard [1]. This novel approach to scheduling allows users to choose when they are available to answer questions or allow sensor collections, increasing their attention and, consequently, improving the quality of their responses. Here we use three BNFs to show the definition of our scheduling language, shown as below.

- *General scheduling as a calendar:*

```
<User>                 :=   <{Calendar}>;
<Calendar>             :=   <calendarid, version,
                            {Context collection}>;
<Context collection>   :=   Question collection | Sensor
                            collection ;
```

Where: we set users as a part of our scheduling language, each user can have multiple calendars for different purpose. According to the iCalendar standard, each calendar

[1] https://www.ietf.org/rfc/rfc5545.txt

can contain multiple context collection events. We collect context data through two ways, questions and sensors.

- *Specific scheduling as question collections:*

<Question collection>	:=	<Qid, dtstart, dtend, status, RRule, questionnaire> ;
<RRule>	:=	<Interval, Count, Frequency> ;
<Frequency>	:=	Daily | Weekly | Monthly | Yearly ;
<questionnaire>	:=	<Questionnaire id, Questionnaire type, Questionnaire category, question relation, {question}> ;
<QuestionnaireType>	:=	Synchronic | Diachronic;
<QuestionnaireCategory>	:=	Spatial context | Event context | Social context | Internal context;
<question>	:=	<question id, content, question type> ;
<question type>	:=	Dichotomous Question | Multiple choice question | Image Question | Free Text Question;

Where: we handle question collection as an event in calendars, with each question collection composed of a Qid to indentify the questions, a Recurrence Rule (RRule), a questionnaire, a status, and a time range specified by a start and end time. The status field indicates whether the event has been accepted by the user, with a value of 1 indicating acceptance and 0 indicating rejection. The Recurrence Rule (RRule) specifies the values used to determine each recurrence and how the event should be repeated. The RRule includes three entity types: interval, count, and frequency. The interval specifies the number of frequency units that must elapse before the next occurrence of the event. The frequency defines the unit of time for repetition, as daily, weekly, monthly, or yearly. The count specifies the number of times the event will be repeated. A questionnaire is composed of multiple questions, a questionnaire_type, and a questionnaire_category. These last two properties are used to explain and differentiate between different questionnaires to users. And for the definition of question_types, we follow the inspiration in [17].

To illustrate our scheduling language in question collection, we present a motivating example: *To understand the user's activity, we choose to use a diachronic questionnaire, with the questionnaire_category belonging to event context. To allow the user greater freedom in their response, we set the type of question to be free text. We plan to send the question to the user five times a day for ten days, by setting an RRule with a frequency of daily, an interval of 10, and a count of 5.*

- *Specific scheduling as sensor collections:*

<Sensor Collection>	:=	<Sid, dtstart, dtend, sensorRRule, sensor> ;
<sensorRRule>	:=	<Sinterval, Scount, Sfrequency> ;
<Sfrequency>	:=	Secondly | Minutely | Hourly ;
<sensor>	:=	<Name, Description, Sensor_type> ;
<Sensor_type>	:=	Social | Motion | Location | Inertial | Device | Ambient;

Where: we handle sensor collection as an event in calendars, with each sensor collection identified by a unique Sid. Since the iCalendar standard's RRule cannot handle recurring events at the minute or second level for sensor collection, we have defined a new entity type, the sensorRRule, which is similar to the RRule in an event but can handle recurrence at the secondly, minutely, and hourly level. Sensor collection is specified by sensor name, description, and sensor_type, which are also used to inform the user of the reason for collecting this type of sensor data. To illustrate our scheduling language as sensor collection, we present another motivating example: *To understand the user's social context, we collect data from Bluetooth devices, which under the Social type in our sensor_type classification. We plan to collect Bluetooth information every minute for ten days, by setting a sensorRRule with a frequency of minutely, an interval of 1, and a count of 14400.*

4. Case Study

To validate the flexibility of our scheduling language for researchers, as described above, we applied it to represent different types of events for collecting context data (sensor collection and question collection) and time recurrence rules in the WeNet program [2]. This allowed us to demonstrate the effectiveness of our approach in a real-world scenario.

In the WeNet experiment, we use iLog to collect data, this application has already been used to collect a significant amount of data in several experiments [15, 18, 19]. In the WeNet experiment, we administered both synchronic (close-ended questionnaires, like the questions

[2] See https://www.internetofus.eu/ for a detailed description of the project plus the possibility of downloading the dataset.

Question	Estimated frequency	Question type	Answer
What are you doing?	Once every 60 minutes	Multiple choice question	providing the participant with 34 categories such as eating, etc.
Where are you?	Once every 60 minutes	Multiple choice question	providing the participant with 26 categories such as home, etc.
Who is with you?	Once every 60 minutes	Multiple choice question	providing the participant with 8 categories such as alone, etc.
What is your mood?	Once every 120 minutes	Image Question	providing the participant with a scale of 5 levels ranging from happy to sad.
What is your birth year?	One time	Free Text Question	the user enters an integer to determine her year of birth.

Figure 2: The example of using questions to collect context data.

1	Name	Type	frequency	Description
2	Bluetooth normal event	social	Once every minute	Bluetooth LowEnergy devices detected
3	Cellular network	device	Once every minute	Cellular network information (cellid, dbm, type)
4	Wifi networks event	ambient	Once every minute	List of Wi-Fi networks detected by the smartphone
5	Ambient temperature event	ambient	Up to 20 times per second	Ambient air temperature (Celsius).
6	Accelerometer event	inertial	Up to 20 times per second	3D vector of the acceleration.
7	Step counter event	motion	Up to 20 times per second	Android API that generates an event when a step is detected
8	locationeventpertime	location	Once every 5 minutes	Location information using GPS connections. (table location)

Figure 3: The example of using sensors to collect context data.

about demographic context, *e.g., What is your birth year? Which gender were you born?*) and diachronic (time diaries questionnaires, like the questions about annotation context, *e.g., What are you doing? Where are you?,* and sensor data, *e.g., Bluetooth, Accelerometer*) collections to participants. In a diary study, data is collected in the form of self-reported activity sequences over time. This type of data is typically collected through self-completed responses that allow individuals to record their activities at fixed intervals. In total, we collected over 110,000 answers from 318 student participants over a period of 28 days. Moreover, the data collection process followed a protocol compliant with the EU General Data Protection Regulation (GDPR).

As shown in Fig. 2, our scheduling language provides researchers with greater flexibility in designing questions with different frequencies using the RRule. In contrast, previous research required researchers to design the frequency of each question individually, as there was no general methodology for generating recurrence rules. Additionally, our scheduling language allows researchers to design questions in their preferred manner, such as free text or multiple choice. Sensor data is automatically collected in the background without requiring user intervention, from a pre-selected list of 56 sensors categorized into six different types, as a simple example illustrated in Fig. 3. Researchers are also given the flexibility to design the recurrence for sensor data collection and classify sensors according to their intended use.

Furthermore, we enable participants to monitor the feedback time through their research calendars. Although we have enhanced the iCal standard, our new scheduling language remains compatible with various open-source packages that visualize calendars. For example, SmartGWT[3] is an open-source calendar that can visualize events. This allows researchers to easily integrate our scheduling language with existing tools and systems to improve the user experience and enhance the quality of context information. The calendar allows users to align data collection events with their real-life schedules and assess whether these events interfere with their daily activities. In the future, we plan to provide users with the ability to modify the timing of their responses to better suit their preferences, and to improve the quality of their responses. There are several papers demonstrate how this data can be used. For example, [20] uses four contextual dimensions data to predict individuals' behaviors. This shows that our scheduling language is useful for both collecting data and future data analysis.

[3]https://smartclient.com

5. Conclusion

This paper proposes a language for modeling the context of data collecting and demonstrates its use in providing a knowledge-level representation of data collected from real experiments on time context. The language includes different types of context collections (question collection and sensor collection) and time recurrence rules. The BNF, defined based on the scheduling language, can be used to collect data from the sensors and self-reports of smartphones. Our case study shows that the scheduling language can be applied in real experiments.

Acknowledgments

The research by Fausto is funded by the European Union's Horizon 2020 FET Proactive project "WeNet – The Internet of us", grant agreement No 823783. The work by Haonan receives funding from the China Scholarships Council (No.202107820038).

References

[1] C. Hoareau, I. Satoh, Modeling and processing information for context-aware computing: A survey., New Generation Computing 27 (2009).

[2] F. Giunchiglia, X. Li, M. Busso, M. Rodas-Britez, A context model for personal data streams, in: Web and Big Data: 6th International Joint Conference, APWeb-WAIM 2022, Nanjing, China, November 25–27, 2022, Proceedings, Part I, Springer, 2023, pp. 37–44.

[3] X. Li, M. Rodas-Britez, M. Busso, F. Giunchiglia, Representing habits as streams of situational contexts, in: Advanced Information Systems Engineering Workshops: CAiSE 2022 International Workshops, Leuven, Belgium, June 6–10, 2022, Proceedings, Springer, 2022, pp. 86–92.

[4] R. Wang, F. Chen, Z. Chen, T. Li, G. Harari, S. Tignor, X. Zhou, D. Ben-Zeev, A. T. Campbell, Studentlife: assessing mental health, academic performance and behavioral trends of college students using smartphones, in: Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing, 2014, pp. 3–14.

[5] Y. Huang, H. Xiong, K. Leach, Y. Zhang, P. Chow, K. Fua, B. A. Teachman, L. E. Barnes, Assessing social anxiety using gps trajectories and point-of-interest data, in: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, 2016, pp. 898–903.

[6] R. Larson, M. Csikszentmihalyi, The experience sampling method, in: Flow and the foundations of positive psychology, Springer, 2014, pp. 21–34.

[7] J. D. Runyan, T. A. Steenbergh, C. Bainbridge, D. A. Daugherty, L. Oke, B. N. Fry, A smartphone ecological momentary assessment/intervention “app” for collecting real-time data and promoting self-awareness, PloS one 8 (2013) e71325.

[8] S. S. Intille, J. Rondoni, C. Kukla, I. Ancona, L. Bao, A context-aware experience sampling tool, in: CHI’03 extended abstracts on Human factors in computing systems, 2003, pp. 972–973.

[9] M. Porta, A dictionary of epidemiology, Oxford university press, 2014.

[10] M. Boukhechba, L. Cai, P. I. Chow, K. Fua, M. S. Gerber, B. A. Teachman, L. E. Barnes, Contextual analysis to understand compliance with smartphone-based ecological momentary assessment, in: Proceedings of the 12th EAI International Conference on Pervasive Computing Technologies for Healthcare, 2018, pp. 232–238.

[11] N. van Berkel, J. Goncalves, S. Hosio, Z. Sarsenbayeva, E. Velloso, V. Kostakos, Overcoming compliance bias in self-report studies: A cross-study analysis, International Journal of Human-Computer Studies 134 (2020) 1–12.

[12] J. Sun, M. Rhemtulla, S. Vazire, Eavesdropping on missing data: What are university students doing when they miss experience sampling reports?, Personality and Social Psychology Bulletin 47 (2021) 1535–1549.

[13] N. van Berkel, J. Goncalves, P. Koval, S. Hosio, T. Dingler, D. Ferreira, V. Kostakos, Context-informed scheduling and analysis: improving accuracy of mobile self-reports, in: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, 2019, pp. 1–12.

[14] I. Bison, H. Zhao, Factors impacting the quality of user answers on smartphones, arXiv preprint arXiv:2306.00627 (2023).

[15] F. Giunchiglia, E. Bignotti, M. Zeni, Personal context modelling and annotation, in: 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), IEEE, 2017, pp. 117–122.

[16] F. Giunchiglia, Contextual reasoning, Epistemologia, special issue: I Linguaggi e le Macchine 16 (1993) 345–364.

[17] N. M. Bradburn, S. Sudman, B. Wansink, Asking questions: the definitive guide to questionnaire design–for market research, political polls, and social and health questionnaires, John Wiley & Sons, 2004.

[18] F. Giunchiglia, M. Zeni, E. Gobbi, E. Bignotti, I. Bison, Mobile social media usage and academic performance, Computers in Human Behavior 82 (2018) 177–185.

[19] F. Giunchiglia, M. Zeni, E. Big, Personal context recognition via reliable human-machine collaboration, in: 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), IEEE, 2018, pp. 379–384.

[20] W. Zhang, Q. Shen, S. Teso, B. Lepri, A. Passerini, I. Bison, F. Giunchiglia, Putting human behavior predictability in context, EPJ Data Science 10 (2021) 42.