The Experience Sampling Method and its Tools: A Review for Developers, Study Administrators, and Participants

TOM GROSS, University of Bamberg, Germany
TONY MALZHACKER, University of Bamberg, Germany

The Experience Sampling Method (ESM) is a popular research method found in many fields to gather rich insights into participants' thoughts, emotions, and daily routines near the moment they happen. During the last decade, many technologically advanced ESM tools emerged that combine manually entered self-reports with automatically collected data from device sensors. In fact, it became difficult to keep track of them. We compiled a survey of ESM and its tools addressing technological capabilities for developers, study design opportunities for study administrators, and answering usability for study participants. It comprises data from 30 systems, applications, and toolkits, which are used for ESM studies. We present our results on the current state of the art from these main user perspectives, list general shortcomings, and give recommendations for future ESM tools.

CCS Concepts: • **Human-centered computing~Human computer interaction (HCI)~HCI design and evaluation methods** • Human-centered computing~Ubiquitous and mobile computing~Ubiquitous and mobile devices • General and reference~Document types~Surveys and overviews

Additional Key Words and Phrases: Experience sampling method; ecological momentary assessment; data collection tools; mobile devices; software architecture; review

ACM Reference format:
Tom Gross, Tony Malzhacker. 2023. The Experience Sampling Method and its Tools: A Review for Developers, Study Administrators, and Participants. *Proc. ACM Hum.-Comput. Interact,* 7, EICS, Article 182 (June 2023), 29 pages, https://doi.org/10.1145/3593234

1 INTRODUCTION

The Experience Sampling Method (ESM) triggers study participants multiple times during one or more days to report their momentary personal circumstances and subjective experiences [30]. Dedicated ESM tools allow study administrators to specify the timing of the triggering, and support study participants in answering the respective questions [19]. With the advent of personal digital assistants and smartphones, tools shifted from an analogue approach (employing beepers, paper, and pencil) to digital solutions [11]. Early digital systems relied on users answering questions about their current situation and emotions. Later systems captured quantitative data from the smartphone's built-in sensors [22]. Context-aware experience sampling tools use these sensor data to (1) infer participants' momentary context (e.g., inferring

Author's addresses: Tom Gross, Tony Malzhacker, Human-Computer Interaction Group, University of Bamberg, Bamberg, Germany, hci(at)uni-bamberg.de.

2573-0142/2023/06 – Article#182… $15.00
https://doi.org/10.1145/3593234

their current location and mode of locomotion), and to (2) present specific question items accordingly (e.g., asking for current personal stress levels right after a telephone call was received and answered) [35; 36].

The main advantage of the ESM method is that it reduces response distortion. Response distortion happens when in empirical studies, participants alter their responses consciously or unconsciously, typically when being asked after and outside of specific circumstances [41]. Additionally, sensor data helps to validate and contextualise the participants' answers, and allow for a more advanced question logic which is better tailored to the participant's current context [39].

The main disadvantage of the ESM method is the increased participant burden caused by the many potential notifications participants receive throughout a day to fill in questionnaires, which can lead to stress and even drop out of the study [73].

The application of ESM as a longitudinal research method in the field suits many scientific disciplines that investigate varying parameters of a person's daily life and require ecological validity (e.g., human-computer interaction, psychology, medical science, social science) [39; 64; 66; 67].

There is an excellent body of knowledge on the ESM method and its tools. Several surveys have addressed general properties, advancements, and challenges of systems and tools for ESM [21; 60; 73]. Other publications have investigated the use of ESM method from a researcher's perspective [29; 62], and the effects of ESM method on participants [43; 76]. However, to the best of our knowledge, there is no comprehensive and integrated review of available ESM tools and their use.

1.1 Goal

Our survey sets out to provide a detailed review of current mobile ESM solutions addressing developers, study administrators, and study participants. According to the editors of last year's EICS 2022 conference— Kris Luyten, Philippe Palanque, Aaron Quigley, and Marco Winckler— the EICS 'community was the first to organise a scientific gathering to foster and exchange research ideas and contributions on how to engineer the effective interactive aspects of a computing system.' [46, p. 1]. They also point out that 'work presented at EICS covers all stages of the engineering life-cycle of interactive systems' [46, p. 2]. From our perspective, properly engineering an ESM tool requires to address the whole life cycle—particularly in the design and development phase which typically involves the developers, in the deployment and configuration phase which typically involves the ESM researchers, up to the phase of maintenance and actual use in the field by end-users. The editorial of EICS 2011 goes in the same direction and emphasises that 'building interactive systems is a multifaceted and challenging activity, involving a plethora of different actors and roles' [23, p. 2]. We compiled a technical overview of commonly supported feature sets of ESM tools and the possibilities and challenges for developers, study administrators, and study participants.

1.2 Method

This survey comprises both free and commercial tools found in scientific literature, peer suggestions, and online search; the latter source was mainly used to identify commercial tools. It integrates our findings with existing literature reviews [e.g., 60; 62; 73]. Specifically—inspired by the 'The PRISMA 2020 Statement: An Updated Guideline for Reporting Systematic Reviews' [58]—we did the following:

We consulted the scientific libraries Web of Science [12], IEEE Xplore [33], ACM Digital Library [1], and Google Scholar [26]

We used the search terms "experience sampling method", "ecological momentary assessment", "ambulatory assessment", and "in situ participatory sensing" and combined them with "tool", "application", and "system"

We concluded a list of 30 systems, applications, and toolkits that were used for ESM studies or in general in an in-situ self-reported or participatory-sensed context

1.3 Contribution

Summarising, in this paper, we make the following main contributions:

1. From a developer's perspective, we provide an overview of the fundamental underlying system architectures, programming paradigms, and technologies ESM tools are based upon and discuss their strength and weaknesses
2. From a study administrator's perspective, we collected and categorised important ESM concepts and properties and to what extend they are supported by ESM tools
3. From a participant's perspective, we present how ESM tools employ facilities to improve usability, user experience to counteract participant burden and enable privacy protection mechanisms
4. And most importantly: we provide those three parts in an integrated manner, since from our experience the interests and requirements of those three groups typically strongly overlap and all need to be considered when it comes to requirements and requirements engineering of ESM tools

1.4 Structure of this Review

Section 2 covers the technical underpinnings of ESM tools. Here, ESM tool developers that are new to this field gain a detailed overview of best practices, real-world implementations and their specific strength and weaknesses. Relevant concepts and features for conducting ESM research are presented in Section 3. It aims to support study administrators in their decision process in what tool to choose for a given study design. Section 4 addresses how these tools support participants in ongoing studies. In Section 5 we provide ideas for future systems by first discussing the current shortcomings of ESM tools. Based on these findings, we then derive recommendations and guidelines for future systems. In Section 6 we finally conclude our review and discuss the limitations of our work.

2 RELATED WORK

Great reviews on the design, implementation, and use of ESM tools have been published—they address technical aspects, configuration aspects, and use aspects.

- *Technical aspects.* Van Bekel et al. [74] analysed 13 ESM tools with respect to their technical features for sensor logging, triggering of questionnaires, branching, availability for operating systems, and configuration capabilities. Pejovic et al. [59] reviewed 12 ESM tools and their technical properties such as the platform they are running on, survey support, and context sensing capabilities. Fischer [21] had a very early and concise review of ESM tools and gave some design recommendations such as 'think client-server' and 'design for authoring'.
- *Configuration aspects.* Rough and Quigley [62] contrasted empirical data on the use of ESM tools by study administrators with the capabilities of 11 ESM tools. They came up

with insightful design recommendations—particularly with respect to end-user development options for study administrators. Harari et al. [29] provide a great collection of smartphone sensors to automatically collect behavioural data in psychology and discuss parameters for sensing studies from a study administrators' perspective (e.g., study duration, sampling rate).
- *Use aspects*. Wenz et al. [76] looked at ESM tool functionality from an end-users—that is, study participants'—perspective. In particular, they investigated to impact of personalised feedback during the study on the participants and their positive reactions. Lawes [43] examined the effect of different modes of contacting study participants on their participation in ESM studies.
- *Combining aspects*. Some work has brought together different perspectives, but not for ESM. At EICS, Hall et al. [28] analysed adaptivity in ubiquitous computing software from the perspective of the developer, the user, and the evaluator. Also at EICS, Batalas and Markopoulos [7] considered requirements for researchers, for participants, and for developers, as input to build their data collection tool.

We benefited from the perspectives and criteria of those reviews and beyond. Nevertheless, to the best of our knowledge, there is no comprehensive and integrated review of available ESM tools and their use from multiple perspectives.

3 ESM TOOLS FROM A DEVELPER'S PERSPECTIVE

This section provides tool developers with the technological state-of-the-art. We aim to provide programmers with an understanding of the state-of-the-art ecosystem of modern ESM tools, including their intricacies and relevant technological factors commonly found in this domain. We argue that a fundamental understanding of the technical underpinnings of these software systems is required to better gauge what current and future software systems are capable of achieving in this space, and also what limitation are imposed by these architectures. This can be valuable (1) when deciding to build new systems from scratch as it provides guides and patterns on how to build them; and (2) when current systems that have to be adjusted or extended with new functionalities. We combined dimensions from the related work above (cf. technical aspects) with dimensions that emerged as relevant from our own work on ESM tools. We used the following six dimensions in our technical analysis: open source access; programming technologies; software architecture; data exchange formats; client platforms; and background processes.

3.1 Open Source Access

More than half of the tools had an open source licence, and most tools with freely available source code on GitHub [49]. This makes code inspection convenient, as whole repositories (including branches, histories, and bug-trackers) can be viewed in the browser without having to download the sources beforehand.

3.2 Programming Technologies

Software systems rarely resemble monolithic structures that are built from scratch every time anew. They are comprised of many individual modules and components, which often incorporate other utility software components or libraries (e.g., third-party libraries for data transmission or dedicated data storage solutions) to help achieve the overarching goal of the system in an orchestrated way.

Programming Languages
ESM tools are built upon a broad spectrum of programming languages: they range from (statically) typed languages (e.g., Java, Swift, or C#) mainly used for native client and server software development, to interpreted solutions (e.g., PHP, Python, or JavaScript) mainly used for server-side software or hybrid client-side approaches with multi-platform support.

Based on these findings, it is apparent that developers should be familiar with at least one set of programming languages that are used for native client development and also one server-side language for central authoring and data storage purposes.

Third-Party Libraries
ESM tools are often based on or integrate third-party toolkits or libraries that handle various tasks, like providing support for sophisticated statistical functionalities [57], enabling cross-platform development [25], simplifying event processing [16], integration of scripting languages [52], easing the process of building server-based web applications [13], providing modular survey functionalities and work flows [3], or helping with integration and processing of device sensor data [42].

These findings show that many of the typical tasks in this domain can be handed over to pre-built software components, which may drastically reduce the development times and efforts needed when this has to be done from scratch. An additional benefit is that these external components were usually tested and evaluated in other real-world scenarios, which may further improve the software reliability and correctness of the ESM solution.

Data Storage Solutions
All ESM tools provide a dedicated data storage solution. On the server side, relational (enterprise) data management solutions (e.g., MS SQL Server, Dynamo DB, or MySQL) were used and sometimes document-based databases like MongoDB. On the client side, the most employed storage solution for persisting user data was SQLite, as it is supported by both Android and iOS, and is even available in modern web browsers. SQL Server 2005 Mobile Edition (SQL Mobile) was also used by one ESM tool [22].

These results show that ESM tools follow standard general practices by delegating data persistence to dedicated software components, which can be queried by standardised APIs and data query languages (i.e. SQL).

3.3 Software Architecture

Software architectures are fundamental to software design principles that permeate all areas of a software system [69]. Here, developers have to make a careful decision about which architectural style to choose for both the data-collecting clients and also for the overall ESM software solution [21]. As modern ESM solutions typically involve at least one dedicated client application to capture participant data and often also an external service for convenient study configuration and data storage, it is crucial for developers to gain insight into both the overall system architecture of all integrated parts of the ESM service and the architecture of the client software itself.

System Architecture
Most ESM tools have a client-server system architecture, where the client is an app running on a mobile device (Android or iOS smartphone). The central server is a computer used (1) as a

centralised data repository for managing and deploying ESM studies via a dedicated graphical user interface (GUI), and (2) as central data storage for study results that are sent back by the clients. Only one of the reviewed tools requires a permanent online connection, as it relies on third-party online questionnaires for participant prompting [31].

The advantages of the client-server model are that studies can be remotely deployed onto the participants' mobile devices, so that they can be monitored in real time, and that study results can be remotely collected without requiring participants to send in their devices.

The disadvantages are increased configuration and maintenance efforts, and costs of additional services hosted on remote machine mostly in the cloud with the risk of security and privacy issues and violations, when storing personal data remotely on such machines, especially if they are self-hosted and not provided by a certified commercial supplier.

Additionally, some ESM tools facilitate hybrid models supporting both architectural styles [22; 32]. Here, study administrators are flexible and can decide for each study, which approach they want to follow (e.g., trading convenience features for privacy measures, if highly personal data is to be collected).

Overall, the client-server model has been and still is the preferred architectural style of ESM tools [21]. This also implies that ESM system developers should be familiar with distributed system architectures and their concepts; and especially with the main properties and issues that are inherent to this architecture, like the need for secure data transmission and consistent data storage in an asynchronous data processing environment.

Client Architecture

Most mobile clients build upon platform-specific SDKs and programming languages such as Java or Swift with native compiler support. The main advantage is that developers can leverage the device's hardware (e.g., gain access to the device's hardware and software sensors and require less main memory and battery power). The downside is that programmers need to know platform specifics (APIs, SDKs, configuration, and programming languages) and client applications are implemented separately for each platform, because binaries cannot be shared.

In order to alleviate this issue, web technologies support cross-platform clients. Here, user interface (UI) elements are implemented by using Hyper Text Markup Language (HTML) and Cascading Style Sheets (CSS), and the domain logic is written in JavaScript. The client application is run and rendered in a web browser window (behaving like a native application), or embedded in a native web container, which behaves like a sandboxed web browser without the encompassing chrome. This web container is thinly wrapped by a native application layer, which might also provide an additional JavaScript-to-native bridge, to access device sensors from within the browser's sandboxed execution environment. The main advantage of this approach is that multiple client platforms can be targeted with only one code base, theoretically simplifying the development and maintenance part of such applications. Another advantage is that core ESM logic can be dynamically updated at any time, since it is interpreted by the web container (i.e., browser) and does not require static re-compilation, deployment, and application approval processes.

On the other hand, the downsides are that programmers also have to be proficient in web technologies, besides knowing the native platform APIs, especially if sensors have to be accessed. These web-based applications require more runtime resources, which leads to increased battery consumption. And due to their interpreted and dynamic nature, these applications may also respond not as quickly to user input [6], when compared to true native solutions, which can hamper usability, especially in the context of ESM, where user prompts should be answered quickly and effortlessly, without requiring much attention from the participant.

The third approach is to use a dedicated cross-platform development toolkit (e.g., Xamarin, or Flutter), which allows for a unified development process and promises to achieve native application performance [25; 50]. The main advantages of this approach— compared to the web container approach—are native performance (and minimal battery impact), native look and feel, and direct sensor access. The downside is that the client application has to be recompiled, if there were fundamental changes in domain logic, and thus entailing a separate deployment and application approval process.

Most open-source solutions follow an event-driven architecture, using either native events provided by the platform runtime, or the browser's event system, if they use a web-container-like approach. This is sometimes accompanied by unifying input sources as data collectors, where user input acquired from prompts and passively collected data from sensors are treated equally—here the participant is just another human sensor [18]. Combined with the aforementioned event-driven client architecture this allows for a flexible and extensible approach, where additional custom sensors or prompt types can be added via plugins, using the same abstract interface as a basis. This part of the system is usually modelled after a general Observer Pattern, notifying subscribed observers only if the state of an input element or that of a sensor has changed [24]. This also enables both pull and push configurations from the perspective of the observer (i.e., the entity being interested in data changes of the respective input sources), which significantly reduces dependencies between software modules.

3.4 Data Exchange Formats

Data exchange over an external channel (network layer, or serial bus) is an integral part of these systems. Data consist of a detailed configuration of a study with all its parameters (e.g., start and end times), triggers used, questionnaires and sensor types typically authored on a central server component and then distributed to the client devices. Even client-only solutions often require configuration files that need to be transferred to the client device (usually via USB-link) before a study can be run [9; 22; 32].

Here, the most common exchange format is JSON, followed by some instances of XML, and two custom file formats [9; 32].

Overall, most of the solutions opt for a text-based, human-readable open format, for which there is a plethora of support in most modern programming languages, boosting cross-programming-language integration. Disadvantages of text-based exchange formats typically involve less efficient data transmission, since the ratio of payload vs. metadata overhead is suboptimal when compared to more densely packed binary formats, like for example, Google Protocol Buffers [27].

3.5 Client Platforms

Most of the analysed ESM tools feature a dedicated Android client, with some also additionally supporting iOS devices. Hybrid approaches can also be found—using either web containers (and native bridges) or dedicated cross-platform development toolkits that reduce the necessary effort in programming specific features twice [68]. There are two solutions that only support iOS [15; 32] and one tool that runs on the deprecated Windows Mobile platform [22].

To enable context-aware experience sampling studies, hardware and software sensors on mobile phones can be accessed via native APIs provided by the respective operating system providers and maintainers (i.e., Google and Apple). Unfortunately, in multi-platform scenarios where devices are not restricted to one specific platform and hardware manufacturer, only a tiny subset of all available sensors can be used for context capture.

A drawback is that the two platforms—Android, and iOS—use different programming languages, SDKs, application architectures, and runtimes. These structural and conceptual incompatibilities require major code rewrites and refactoring. This problem can be approached by hybrid development frameworks like PhoneGap/Apache Cordova [2], which employ web technologies (HTML5, CSS, JavaScript) to facilitate a unified development experience. Newer approaches like Flutter [25] use system-level technology, a dedicated programming language (i.e., Dart), and an application framework, which compiles to native code for both platforms simultaneously.

3.6 Background Processes

ESM-tools developers also face the problem that platform-specific APIs—especially those required for background processing and sensor access—quickly evolve between operating system (OS) releases, also leading to incompatibilities within the same platform environment. For example, new privacy features and battery-saving measures on Android led to many applications breaking or being rendered completely useless for an ESM scenario [20]. This leads to additional effort for developers of having to maintain and update these tools for new OS releases constantly. Some tried to remedy this issue by providing dedicated locked-down, study-administrator-owned devices, including a pre-configured and locked OS that cannot be further updated or altered by participants [47].

Overall, ESM tools on smartphones rely on the proper functioning of the ESM app running as a background process over an extended period of time in which users typically want and need to use other applications in the foreground from time to time. Yet, proper engineering of background processes is becoming increasingly complex, since battery saving features have become part of smartphone firmware. Online collections and forums [e.g., 72] have emerged that bring together information on different smartphone hardware types and their operating systems and firmware and how they deal with background processes. The Google Doze mode that has been available since Android version 6 is often not properly implemented by the different vendors. Currently, Samsung, OnePlus, Huawei, and Xiaomi are the top three scorers in making background process developers' life hard [e.g., 72].

3.7 Summary

Many tools are open-source and present a valuable resource for developers, as they offer valuable insights, including source-code-level documentation with detailed implementations, guides, and best practices. Table 1 in Appendix A.1 gives an overview of the technical aspects of the reviewed ESM tools.

There are no clear trends in programming technologies, except for native client development, where the respective proprietary software development technologies are used (i.e., Java on Android, and Swift with Objective-C on iOS).

The most common architectural style is a client-server approach combined with native clients. Hybrid client approaches are not common in this domain, although advertised as user-friendly time savers in the development process [2; 25; 50].

Configuration data is typically encoded using a textual format. Most supported formats are JSON and XML, with some tools using a custom text-based data format.

The most supported client platform is Android, followed by iOS (and one tool using the deprecated Windows Mobile platform).

From a developers' perspective, ESM tools often resemble modern client-server stacks, consisting of a central data repository and dedicated mobile clients, with web technologies and standards for communication (e.g., REST, HTTP and JSON).

4 ESM TOOLS FROM A STUDY ADMINISTRATOR'S PERSPECTIVE

This section focuses on the fundamental properties inherent to the method of experience sampling and how they are supported by current ESM tools. Here also we combined dimensions from the related work above (cf. configuration aspects) with dimensions that emerged as relevant from our own work on ESM tools. We, therefore, focus on probe types, trigger types, and study authoring facilities.

4.1 Probe Types

Probes can be split into: user probes that actively ask participants to fill in data (also called prompts, questions, or items); and sensor probes that passively capture sensor data in the background.

User Probes

The most common user probe types are: text, radio, checkbox, Likert scale, slider, photo and video, audio, and affect grid.

User probe types that were supported by all tools are single choice (radio), multiple choice (checkbox), free text (text input), and display text (no user input). Sometimes single and multiple-choice options are paired with an optional free text input, called 'other' option. Unsurprisingly, these probe types are common input types that have been available since the early days of GUIs.

Most tools also supported slider/range inputs, Likert/scale inputs, audio/voice input, image/photo input, date/time input fields, and (geo-)location pickers. Here, slider/range-type and Likert/scale-type prompts are most represented. A clear separation between those types was not always feasible because some tools intermingled discrete scales (Likert) and continuous scales (range/slider). Both input types provide an upper and lower limit. Likert scales incorporate discrete answer options (usually five or seven) and typically ask for participant agreement. Multimedia input is also widely supported by most ESM tools, as well as inputs for temporal and spatial data (i.e., date and time inputs and location pickers).

Few ESM tools feature video input, file input/upload, rating (stars/pictorial) scales, ranking (ordering), (affect) grid input, table/matrix input, annotate/highlight/mark image, or a URL widget. This is surprising for two-dimensional-selection prompts (affect grids), which were declared to be one of the most important types for ESM [73], but were only featured by three tools [32; 40; 47].

Rare user probe types supported by at most one tool are amount-of-time input, acknowledge/confirm, captcha verification prompt, colour picker, combination choice slider, constant sum, draw widget, drill down widget, heart rate widget, heat map, hot spot, meta-information question, percentage and percentage fill input, pick, group, and rank widget, photographic effect meter, quick answer, self-assessment manikin widget, screen capture, side-by-side widget, signature input, timing input, working memory task widget.

Sensor Probes

Sensor probes play an essential role in collecting contextual data and inferring concepts such as the current activity (e.g., sitting, running), environment (e.g., work, home), or availability (e.g., in a meeting, answering a phone call). Sensor probes can be beneficial for end users, because (1) they reduce participant burden, since relevant information can be inferred automatically without requiring active participants' input; (2) they help validate self-reported participant data as a second objective input source; and (3) they facilitate event-contingent notifications and thus allow specific questions regarding the participant's current situation.

Due to the increasing importance of protecting users' privacy and the reduction of battery drain caused by excessively querying sensor data, systems have more and more limitations for capturing data. This may explain the lack of sensor probe support by commercial ESM tools, which typically only support location information—provided by the Global Positioning System (GPS) sensor—with an option to integrate more sensors after further consultation or offering custom-built sensor devices [51]. Providing specific devices for sensor data is offered by one non-commercial tool [47]. The tool provides its own pre-configured, pre-tuned, and locked Android smartphones.

Sensor probes are hardware- or software-based. The most supported hardware-based sensor was GPS-based location. Even modern browsers are granting access to a device's current location via Geolocation API [75]. Common software-based sensors are application usage and statistics (start and stop, installation), system/device status and statistics (phone on and off, a user interacting, display on and off), call/SMS (incoming and outgoing, logs), and calendar access.

4.2 Trigger Types

Three types of triggers initiate reminders (usually auditive and visual system notifications) for participants to fill in data: interval trigger, signal trigger, and event trigger. Most widely supported is the interval trigger, which is executed on a fixed schedule (e.g., every evening at 20:00h). Signal-contingent triggers fire randomly within a pre-defined interval—typically with 8-10 intervals with 1 prompt each per day [73]. Additionally, a timespan can be specified until the next signal is allowed (e.g., 15 minutes after the last signal). Event-contingent triggers are not as widely supported. These events are provided by hardware and software sensors inferring context and higher concepts or initiated by the participants themselves before or after a study relevant event has taken place.

4.3 Study Authoring Facilities

Study authoring facilities can be grouped into features aiding in study design, study management, and data management.

To help study administrators in the study design phase: GUIs support designing forms and configuring triggers, form preview and test mode, prompt order randomisation, answer option randomisation (single and multiple-choice types), question libraries, and study schedule editors.

The most common features provide a GUI for creating forms and triggers, including drag & drop functionalities for quick question reordering. One system [61] also offers several validation and 'expert' review features for created forms, offering concrete suggestions for improvement.

Randomisation support for both prompt order and answer option order (for choice-type prompts) can be helpful to break up established or learned answering routines, and thus increase data quality for longitudinal studies.

Question libraries are rarely supported. Pre-built forms (including standardised scales) can help study administrators to get studies up and running quickly. The same holds for a graphical schedule editor (e.g., in calendar format), where researchers can plan studies ahead of time in a visual and more familiar manner than just entering start and end times in plain input fields.

Supporting study administrators in managing ongoing studies is also essential. This is facilitated by live study statistics and insights (dashboards), team management (groups, rights, and roles), discussion boards and collaboration features, participant group randomisation, and dedicated participant contact channels.

For an understanding of the state of ongoing studies, many client-server-based tools provide a statistical overview of key metrics (dashboards), like the number of prompts sent and missed or participant completion rate and compliance. Sometimes these statistical overviews can also be directly shared with participants [40].

Managing rights and roles of co-administrators is a crucial task supported by some tools, as well as providing collaboration features (e.g., like discussion boards, artefact linking facilities). There are features geared towards managing participants, like automatically and randomly selecting participants to (pre-defined) groups and a dedicated feedback channel to quickly contact enrolled participants via chat or email.

After a study ends, participant data is collected, cleaned, and statistically evaluated. To ease this process, ESM tools provide automatic data transfer facilities from clients to a central repository (on client-server architectures), Comma Separated Values (CSV) exports, and dedicated data management and data browsing facilities. The collection of study results is simplified, where data is automatically (or manually triggered by the participant) transferred to a central data repository, from which it can further be processed and exported, without requiring the devices to being sent in first and then physically connecting to each of them via cable to get the data off the device. All tools support exports in CSV format. Although some tools also provide simple statistical functionality within the application itself, sophisticated data analysis is expected to be done within dedicated statistic applications. Some ESM tools provide built-in scripts and functionalities to use these statistical tools within the ESM application itself [4] or scripts for simplifying the export process [8].

Data cleaning within the ESM tool is rarely supported; we only found Teamscope [70] to provide a dedicated GUI-based query manager for resolving data inconsistencies.

4.4 Summary

Most ESM tools help study administrators in setting up and running ESM studies with a plethora of dedicated functionalities and convenience features. There are dedicated tools that provide a range of sensor types, and come with many domain-specific question types. The largest set of functionalities is provided by commercial ESM solutions, which typically also come with extensive end-user documentation and video tutorials, further assisting even less-technical-versed study administrators in running sophisticated study designs.

Table 2 in Appendix A.2 provides details on study-administrator-related aspects of the tools we reviewed.

5 ESM TOOLS FROM A PARTICIPANT'S PERSPECTIVE

We used the following two major dimensions in our analysis: usability and user experience, and privacy.

5.1 Usability and User Experience

Usability is concerned with the effectiveness, efficiency, and the satisfaction a computer systems provides for a specific user in a given context to achieve certain goals [37]. The facilities in this category have been collected from a plethora of single papers that typically identified one or more of those features (cf. subsection on Method above). They include offline-usage modes, customisable study start and end times, questionnaire preview and test modes, support channel and helpdesk facilities, participant-initiated events, comparison of own responses vs. baseline response, provision of study results and insights after study ended, list of available studies and self-registration, customisation of notification modality (auditive, visual, tactile modes), and mute or snooze buttons.

The most commonly supported feature is dedicated offline modes, with additional options to manually or automatically sync to a cloud-based central repository (on client-server architectures). This simplifies the process, because participants do not worry about network coverage, especially in-door or in rural areas. It can also reduce data plan costs by only syncing data if a Wi-Fi connection is available, which usually has no transfer fees or quota attached to it. It can also increase the battery life of the device, thus requiring fewer recharge cycles during the course of a study.

To support customisation, some tools allow participants to override pre-planned sampling start and end times to better fit the schedule to participants' daily routines.

To provide upfront insights on what will be asked during the study, some tools allow preview modes of the study questionnaires. This may increase familiarity with the tool and question types, and thus further reduce fear of use.

Many tools offered a dedicated help channel to offer quick support, further inducing trust between participants and study administrators.

To increase the participants' motivation, some ESM tools support in-app statistics, where participant responses are scored and compared against baseline scores of other participants; or they simply provide an overview and summary of given responses.

Some ESM tools provide further insights and study outcomes, after a study has concluded. This may positively affect participants to recruit them for future studies, for which some tools also provide easy follow-up sign up functionalities.

Although most devices allow to set the modality of notification system-wide, one tool [47] explicitly allowed participants to specify if they want to get notified by vibration or sound.

Lastly, dedicated mute and snooze buttons are offered by few ESM tools. They are used to delay answering of questionnaires and to quickly stop device notifications altogether [32; 71]. This is helpful in situation where interactions with the client device are socially inappropriate or disturbing for participants to do (e.g., private meetings, public car rides, or emergency situations). They are also required in situations where the participants cannot and should not use a mobile device (e.g., while driving a car).

5.2 Privacy

Privacy issues of ESM systems have already been discussed in its early days [36], with many regional laws regulating and constraining the collection of personal data (e.g., the General Data Protection Regulation (GDPR) by the European Union [17]).

Privacy measures can be actively controlled by participants within the application itself, or automatically enabled by the ESM tool. Actively controllable features include a stop-data-collection button, transparency upon registration (e.g., a list of all data-collecting sensors), an in-app overview of all collected data, and several private modes. Some ESM tools already provide means to stop all automatic data collection processes [8; 45]: prompts are not displayed anymore and the collection of sensor data is shut down for the time being.

Another privacy-related function is to provide a list of all data-collecting sensors upon study registration, with the option for participants to opt out of specific sensor types (e.g., GPS tracking) for the rest of the study.

An in-app overview of all on-device captured data is sometimes provided, increasing data transparency for participants, which may enforce trust in this study and tool. Being transparent about data use is one of the most sought-after and highly valued features for users in this domain [45].

Following the concept of private modes found in many modern web browsers, some tools provide functionality for participants in the context of ESM: they do not track any personal identifiers while this mode is active and simultaneously dilute spatial and temporal resolutions of collected sensor data.

System-wide privacy features that cannot be directly controlled by participants are cryptographic hashing of personal identifiers, on-device encryption of all data, automatic data degradation (omission, randomisation, rounding); and system certifications and privacy-related guarantees. The most common facilities are one-way cryptographic hashes to encrypt personal data, or using on-device encryption mechanisms (usually provided by the client platform) for all collected data sets. This applies to a client-server architecture, where participant data is first encrypted and stored on the device before it is transmitted to a cloud-based repository.

Some ESM tools provide study administrators with automatic anonymisation mechanisms, like omitting data fields, random anchoring of time entries, or rounding numeric values, adjusting sensor polling rates, and thus decreasing temporal and spatial precision of data readings [4; 77].

Lastly, commercial ESM solutions are often advertised as meeting strong privacy and data safety guarantees, by being officially audited and certified by third-party organisations. Examples of such certificates are Health Insurance Portability and Accountability Act (HIPAA) Compliance, GDPR Compliance, Food and Drug Administration (FDA) Compliance, or ISO 27001 Compliance. Unfortunately, such certifications are not found in open-source tools. They sometimes list certified software components, i.e., cloud-based data stores like Amazon AWS or DynamoDB [54; 77], but of course, this does not automatically guarantee data safety for the entire ESM tool and all its components.

5.3 Summary

Most ESM tools supported offline use, which cuts down the costs of mobile data plans and alleviates participants from the burden of constantly checking their connectivity. Further positive findings are tools, which provide dedicated help channels. Here, participants could quickly get in touch with study administrators and many user-controlled customisation options, to better adapt to a participant's current context and likings, which also agrees with common usability best practices for interactive systems [53]. Further details on participant-related aspects of ESM tools can be found in Table 3 in Appendix A.3.

Only seven ESM tools provided system-wide privacy certifications and guarantees out-of-the-box (most of them being commercial solutions). However, many tools employed at least on-device encryption and pseudonymisation methods, provided up-front transparency on which sensors are being used on the device, and included the option to disable unwanted sensor types or to completely shut off data collection altogether.

6 LESSONS LEARNED FOR FUTURE ESM TOOLS

We cover general issues relevant for all stakeholders, as well as dedicated issues found for tool developers, study administrators, and participants.

General issues are missing support to share ESM studies between tools; tools supporting only one platform (i.e., Android); and tools that lack error mitigation and error reporting facilities.

The most crucial issue is that ESM studies—including protocols, forms, triggers, schedule, and sensor configuration—cannot be shared amongst EMS tools. This general lack of interoperability significantly complicates study replication, because study protocols need to be recreated each time anew, whenever a group of ESM researchers has no access to the tool, the tool is no longer available and maintained, or does not support the targeted client platform (version).

Regarding platform support, most ESM tools did not target both popular mobile platforms (i.e., Android and iOS). The lack of cross-platform support may lead to selection bias when conducting experience sampling on participants own devices, systematically excluding certain user groups a priori.

From a study administrator's perspective, another issue is a lack of remote insights into the cause of missing data. Sources of these nonresponses could be participants not answering in inconvenient circumstances or due to increasing response fatigue [10; 65]; or technical issues caused by hardware and software malfunctioning (e.g., devices running out of battery) . To tackle this problem, some tools provide debug modes or log files that can be inspected after the results were transferred back to the study administrator, including time stamps, but these do usually not include runtime errors (including important stack traces of the program itself). To our knowledge, there is only one tool that features sophisticated error analytics support [55]. Not knowing what actually caused an error may also negatively affect the user experience.

Other issues relevant for study administrators are that many open source tools have poor or only technical documentation or require technical expertise; most tools lack question and form libraries; and missing support for study pilot (test) modes.

Many of the free and open-source ESM solutions require tech-savvy study administrators to set up and run. Many open source tools require study administrators to use command line tools, version control systems, and directly edit source code files.

Building questionnaires from scratch can be tedious—having a question library with pre-configured items or even domain-specific questionnaires for later reuse can be helpful for study administrators. Unfortunately, only one tool [61] featured a dedicated question library.

For study administrators, doing pilot tests of a study on an actual device can be helpful in uncovering layout-related and protocol-related issues in advance—especially, if prompts are triggered conditionally based on previous answers or sensor readings. Pilot tests need to be exported to participants' devices (including debug start times and conditions), or to be manually started. This can be a laborious process and could be alleviated by providing preview modes directly within the questionnaire-design UI. Unfortunately, preview modes for questionnaires are not common amongst ESM tools.

From a participant's perspective, the most prominent shortcomings of ESM tools are: (1) lack of multi-language support, (2) non-skippable prompts, and (3) non-customisable font sizes. Many ESM tools do not allow participants to switch languages. Non-skippable prompts may lead to increased participant burden and may finally result in more incomplete data, as participants do not complete questionnaires, or only conveniently. Some study designs might require mandatory prompts participants have to answer (especially on screening and intake forms), but this should not be the default. Prompts should be easy to answer and should be dismissible if inconvenient in the current context.

Font size plays the most important role, besides contrast and font style. Unfortunately, only some tools allow participants to change the font size of prompts, which should be a default feature for all user elements on mobile devices, as they can starkly vary in screen size and resolution.

From the perspective of an ESM tool developer, who wants to use, integrate, or extend open source solutions, the following issues were found: (1) use of inconsistent terminology, (2) use of many fundamental differing programming technologies, (3) lack of complete documentation, and (4) lack of established standards for developing ESM tools.

The main issue with open source tools is their inconsistent terminology with regard to concepts of ESM. Here, classes, object, or entities have vastly differing names although they essentially represent the same concept or even worse, slightly related or unrelated concepts, which further increases confusion among developers trying to understand, port, or extend functionality.

Also, open-source ESM tools use fundamental differing programming technologies, which makes porting and extending parts of these systems difficult for programmers that are not familiar with all technologies involved.

Another obstacle for tool developers is missing or incomplete documentation. They are often forced to dig deep into code to understand how APIs are supposed to work, with only superficial user documentation and minimal hands-on examples. This entails additional effort and time developers have to invest in these tools.

For tool developers, this all cumulates to the general problem, that there appear to be no established standards or guidelines on how to build state-of-the-art ESM applications. From our current point of view, many tools leave the impression that they were built from scratch without integrating best practices or lessons learned from other existing ESM tools, giving them a more one-shot prototype character than a well-thought-out system built to last for several years.

Future ESM tools should also address the lack of interoperability. To our knowledge, there is only one tool that specifically addresses this issue [4]. Regardless of personal preferences and monetary means, everybody in this research community would greatly benefit from easy-shareable research designs. The importance of consistently reporting study metrics has also been stressed [73], which makes replication more feasible and promotes good scientific practices.

7 CONCLUSIONS

In this review, we looked into state-of-the-art ESM tools focusing on the perspective of the three stakeholders: tool developers, study administrators, and participants. We provided detailed overviews and discussions of important aspects from each point of view, listed current shortcomings and provided ideas for future ESM systems.

We see the relevancy of ESM tools and particularly looking at those tools from a developer's, a reseacher's, and a user's perspective in multiple ways, as pointed out in the related work. Furthermore, ESM tools have been successfully used in the domain of engineering interactive systems and so we hope that this paper not only informs EICS researchers interested in development aspects, but also interested in using ESM tools as researchers to capture information in-situ.

Our work is limited by the fact that we did not consider all available tools that can be found in the wild (but we opted to include a balanced ratio of free vs paid solutions). A further limitation of our findings can be attributed to pay walls of commercial ESM tools: we had no access to source code files and thus had to rely on public statements made by companies on what their systems are built on. Thus, we had to rely on statements and illustrations found in related academic papers. In general, we also did not pay for any of the 30 analysed ESM tools, but we tried to register for free (demo) accounts and used freely available documentation and (video) tutorials.

ACKNOWLEDGMENTS

We thank the members of the Cooperative Media Lab of the University Bamberg. We also thank the reviewers for their valuable comments.

REFERENCES

[1] ACM. *ACM Digital Library*. ACM Inc., https://dl.acm.org/, 2022. (Accessed 2/1/2023).

[2] Apache. *Apache Cordova*. https://cordova.apache.org/, 2022. (Accessed 20/04/2021).

[3] Apple. *ResearchKit – Create Powerful Apps for Medical Research*. http://researchkit.org/, 2022. (Accessed 2/1/2023).

[4] Arslan, Ruben C, Walther, Matthias P and Tata, Cyril S. (2020). form: A Study Framework Allowing for Automated Feedback Generation and Complex Longitudinal Experience-Sampling Studies using R. *Behavior Research Methods* 52, 1. pp. 376-387.

[5] Bachmann, Anja, Zetzsche, Robert, Schankin, Andrea, Riedel, Till, Beigl, Michael, Reichert, Markus, Santangelo, Philip and Ebner-Priemer, Ulrich. (2015). ESMAC: A Web-Based Configurator for Context-Aware Experience Sampling Apps in Ambulatory Assessment. In *Proceedings of the 5th EAI International Conference on Wireless Mobile Communication and Healthcare* (London, Great Britain). ICST. pp. 15-18.

[6] Batalas, Nikolaos, aan het Rot, Marije, Khan, Vassilis Javed and Markopoulos, Panos. (2018). Using TEMPEST: End-User Programming of Web-Based Ecological Momentary Assessment Protocols. *Proceedings of the ACM on Human-Computer Interaction* 2, EICS. pp. 1-24.

[7] Batalas, Nikoloas and Markopoulos, Panos. (2012). Considerations for Computerised In Situ Data Collection Platforms. In *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems - EICS 2012* (June 25-26, Copenhagen, Denmark). ACM, N.Y. pp. 231-236.

[8] Behavio. *Funf | Open Sensing Framework*. https://www.funf.org/about.html, 2022. (Accessed 2/1/2023).

[9] BlueJayVentures. *PIEL Survey | Experience Sampling Method (ESM) | Ecological Momentary Assessment (EMA) | Survey Software*. https://pielsurvey.org/, 2022. (Accessed 2/1/2023).

[10] Broda, Michael. (2017). Using Multilevel Models to Explore Predictors of High School Students' Nonresponse in Experience Sampling Method (ESM) Studies. *Social Science Computer Review* 35, 6. pp. 733-750.

[11] Carter, Scott, Mankoff, Jennifer and Heer, Jeffrey. (2007). Momento: Support for Situated Ubicomp Experimentation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA). Association for Computing Machinery. pp. 125-134.

[12] Clarivate. *Web of Science*. https://webofknowledge.com, 2022. (Accessed 2/1/2023).

[13] Django. *The Web Framework for Perfectionists with Deadlines | Django*. https://www.djangoproject.com/, 2022. (Accessed 2/1/2023).

[14] Enketo. *Web Forms Evolved. Deploy and Conduct Surveys that Work without a Connection, on Any Device.* https://enketo.org/, 2022. (Accessed 2/1/2023).

[15] ESMCapture. *Experience Sampling Made Simple*. http://esmcapture.com/, 2022. (Accessed 2/1/2023).

[16] EsperTech. *Complex Event Processing, Streaming Analytics, Streaming SQL - EsperTech*. https://www.espertech.com/, 2022. (Accessed 2/1/2023).

[17] EU. *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC (General Data Protection Regulation).* https://eur-lex.europa.eu/eli/reg/2016/679/oj, 2022.

[18] Evans, Bob. (2016). Paco—Applying Computational Methods to Scale Qualitative Methods. In *Ethnographic Praxis in Industry Conference Proceedings* (Minneapolis, Minnesota, USA). Wiley Online Library. pp. 348-368.

[19] Feldman Barrett, Lisa and Barrett, Daniel. (Summer 2001). An Introduction to Computerised Experience Sampling in Psychology. *Social Science Computer Review* 19, 2. pp. 175-185.

[20] Ferreira, Denzil, Kostakos, Vassilis and Dey, Anind K. (2015). AWARE: Mobile Context Instrumentation Framework. *Frontiers in ICT* 2. pp. 6.

[21] Fischer, Joel E. (2009). Experience-sampling tools: a critical review. *Mobile living labs 09: Methods and tools for evaluation in the wild.* pp. 35.

[22] Froehlich, Jon, Chen, Mike Y, Consolvo, Sunny, Harrison, Beverly and Landay, James A. (2007). MyExperience: A System for In Situ Tracing and Capturing of User Feedback on Mobile Phones. In *Proceedings of the 5th international conference on Mobile systems, applications and services* (San Juan, Puerto Rico). Association for Computing Machinery. pp. 57-70.

[23] Funk, Matthias, Liang, Rong-Hao, Palanque, Philippe, Hu, Jun and Markopoulos, Panos. (June 2021). Engineering Interactive Computing Systems 2022: Editorial Introduction. *Proceedings of the ACM on Human-Computer Interaction* 5, EICS, 193. pp. 193:1-193:4.

[24] Gamma, Erich, Helm, Richard, Johnson, Ralph and Vlissides, John. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley Longman Publishing Co., Inc.,

[25] Google. *Flutter - Beautiful Native Apps in Record Time.* https://flutter.dev/, 2022. (Accessed 2/1/2023).

[26] Google. *Google Scholar.* https://scholar.google.com/, 2022. (Accessed 2/1/2023).

[27] Google. *Protocol Buffers | Google Developers.* https://developers.google.com/protocol-buffers, 2022. (Accessed 2/1/2023).

[28] Hall, Malcolm, Bell, Marek, Morrison, Alistair, Reeves, Stuart, Sherwood, Scott and Chalmers, Matthew. (2009). Adapting Ubicomp Software and its Evaluation. In *Proceedings of the 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems - EICS 2009* (July 15-17, Pittsburgh, PA). ACM, N.Y. pp. 141-148.

[29] Harari, Gabriella M, Lane, Nicholas D, Wang, Rui, Crosier, Benjamin S, Campbell, Andrew T and Gosling, Samuel D. (2016). Using Smartphones to Collect Behavioral Data in Psychological Science: Opportunities, Practical Considerations, and Challenges. *Perspectives on Psychological Science* 11, 6. pp. 838-854.

[30] Hektner, Joel M., Csikszentmihalyi, Mihaly and Schmidt, Jennifer A. (2006). *Experience Sampling Method: Measuring the Quality of Everyday Life.* Sage Publications Inc., Thousand Oaks, CA.

[31] Hofmann, Wilhelm and Patel, Paresh V. (2015). SurveySignal: A Convenient Solution for Experience Sampling Research Using Participants' Own Smartphones *Social Science Computer Review* 33, 2. pp. 235-253.

[32] iDialogPad. *This is an EMA (Ecological Momentary Assessment) and General Questionaire Universal App for iOS.* https://github.com/gemu2015/iDialogPad, 2022. (Accessed 2/1/2023).

[33] IEEE. *IEEE Xplore.* https://ieeexplore.ieee.org, 2022. (Accessed 03/04/2021).

[34] ilumivu. *mEMA App : ilumivu.* https://ilumivu.com/solutions/ecological-momentary-assessment-app/, 2022. (Accessed 2/1/2023).

[35] Intille, Stephen S, Rondoni, John, Kukla, Charles, Ancona, Isabel and Bao, Ling. (2003). A Context-Aware Experience Sampling Tool. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA). Association for Computing Machinery. pp. 972-973.

[36] Intille, Stephen S, Stone, AA and Shiffman, S. (2007). Technological Innovations Enabling Automatic, Context-Sensitive Ecological Momentary Assessment. *The Science of Real-Time Data Capture: Self-Reports in Health Research.* pp. 308-337.

[37] ISO. *Ergonomics of Human-System Interaction — Part 11: Usability: Definitions and Concepts.* https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en, 2018. (Accessed 20/04/2021).

[38] Karr, Chris. *Purple Robot is an Android Application for Creating Automated Experiences on the Android Platform with Sensing and Inference Capabilities.* https://github.com/cbitstech/Purple-Robot, 2022. (Accessed 19/07/2021).

[39] Keusch, Florian and Conrad, Frederick G. (2021). Using Smartphones to Capture and Combine Self-Reports and Passively Measured Behavior in Social Research. *Journal of Survey Statistics and Methodology.*

[40] Koval, Peter, Hinton, J, Dozo, N, Gleeson, John, Alvarez, Mario, Harrison, A, Vu, D, Susanto, R, Jayaputera, G and Sinnott, Richard. *SEMA3: Smartphone Ecological Momentary Assessment, Version 3.* https://sema3.com/, 2019. (Accessed 2/1/2023).

[41] Larson, Reed and Csikszentmihalyi, Mihaly. (2014). *The Experience Sampling Method.* In Csikszentmihalyi, Mihaly, ed. Flow and the Foundations of Positive Psychology. Springer-Verlag, Heidelberg. pp. 21-34.

[42] Lathia, Neal, Rachuri, Kiran, Mascolo, Cecilia and Roussos, George. (2013). Open Source Smartphone Libraries for Computational Social Science. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication* (Zurich, Switzerland). Association for Computing Machinery. pp. 911-920.

[43] Lawes, Mario, Hetschko, Clemens, Sakshaug, Joseph W and Grießemer, Stephan. (2021). Contact Modes and Participation in App-Based Smartphone Surveys: Evidence From a Large-Scale Experiment. *Social Science Computer Review*. pp. 1-17.

[44] LifeData. *LifeData Experience Sampling App | LifeData Experience Sampling App for Research*. https://www.lifedatacorp.com/, 2022. (Accessed 2/1/2023).

[45] Ludwig, Thomas, Dax, Julian, Pipek, Volkmar and Randall, Dave. (2016). Work or Leisure? Designing a User-Centered Approach for Researching Activity "in the Wild". *Personal and Ubiquitous Computing* 20, 4. pp. 487-515.

[46] Luyten, Kris, Palanque, Philippe, Quigley, Aaron and Winckler, Marco. (June 2022). Engineering Interactive Computing Systems 2022: Editorial Introduction. *Proceedings of the ACM on Human-Computer Interaction* 6, EICS, 149. pp. 149:1-149:3.

[47] Meers, Kristof, Dejonckheere, Egon, Kalokerinos, Elise K, Rummens, Koen and Kuppens, Peter. (2020). mobileQ: A Free User-Friendly Application for Collecting Experience Sampling Data. *Behaviour Research Methods* 52, 4. pp. 1510-1515.

[48] Metricwire. *Metricwire Inc. | Real-World Intelligence | Value Based-Decisions*. https://metricwire.com/, 2022. (Accessed 19/07/2021).

[49] Microsoft. *GitHub: Where the World Builds Software · GitHub*. https://github.com/, 2022. (Accessed 2/1/2023).

[50] Microsoft. *Xamarin | Open-Source Mobile App Platform for .NET*. https://dotnet.microsoft.com/apps/xamarin, 2022. (Accessed 2/1/2023).

[51] movisensXS. *movisensXS – Experience Sampling Solution. movisens GmbH*. https://www.movisens.com/en/products/movisensxs/, 2022. (Accessed 2/1/2023).

[52] Mozilla. *MDN Web Docs*. https://developer.mozilla.org/en-US/, 2022. (Accessed 20/04/2021).

[53] Nielson, Jakob. *10 Usability Heuristics for User Interface Design*. https://www.nngroup.com/articles/ten-usability-heuristics/, 1994. (Accessed 20/04/2021).

[54] O'Reilly-Shah, Vikas and Mackey, Sean. (Apr-Jun 2016). Survalytics: An Open-Source Cloud-Integrated Experience Sampling, Survey, and Analytics and Metadata Collection Module for Android Operating System Apps. *JMIR mHealth and uHealth* 4, 2. pp. 17-26.

[55] ODK-X. *Build your Custom Data Management App*. https://odk-x.org/, 2022. (Accessed 2/1/2023).

[56] ODK. *Collect Data Anywhere*. https://getodk.org/index.html, 2022. (Accessed 2/1/2023).

[57] OpenCPU. *OpenCPU - Producing and Reproducing Results*. https://www.opencpu.org/, 2022. (Accessed 2/1/2023).

[58] Page, Matthew J., McKenzie, Joanne E., Bossuyt, Patrick M., Boutron, Isabelle, Hoffmann, Tammy C., Mulrow, Cynthia D., Shamseer, Larissa, Tetzlaff, Jennifer M., Akl, Elie A., Brennan, Sue E., Chou, Roger, Glanville, Julie, Grimshaw, Jeremy M., Hróbjartsson, Asbjørn, Lalu, Manoj M., Li, Tianjing, Loder, Elizabeth W., Mayo-Wilson, Evan, McDonald, Steve, McGuinness, Luke A., Stewart, Lesley A., Thomas, James, Tricco, Andrea C., Welch, Vivian A., Whiting, Penny and Moher, David (2021). The PRISMA 2020 Statement: An Updated Guideline for Reporting Systematic Reviews. *Systematic Reviews* 10, 89. pp. 2-11.

[59] Pejovic, Veliko, Lathia, Neal, Mascolo, Cecilia and Musolesi, Mirco. (2016). *Mobile-Based Experience Sampling for Behaviour Research*. In Tkalcic, Marko, De Carolis, Berardina, de Gemmis, Marco, Odic, Ante and Kosir, Andrej, eds. Emotions and Personality in Personalised Services. Springer-Verlag, Heidelberg. pp. 141-161.

[60] Pejovic, Veljko, Lathia, Neal, Mascolo, Cecilia and Musolesi, Mirco. (2016). *Mobile-Based Experience Sampling for Behaviour Research*. In Emotions and personality in personalised services. Springer. pp. 141-161.

[61] Qualtrics. *Online Survey Software - 1B+ Surveys Annually | Qualtrics*. https://www.qualtrics.com/uk/core-xm/survey-software/, 2022. (Accessed 2/1/2023).

[62] Rough, Daniel J and Quigley, Aaron. (2020). End-User Development of Experience Sampling Smartphone Apps-Recommendations and Requirements. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 2. pp. 1-19.

[63] Rough, Daniel John and Quigley, Aaron John. (2017). Jeeves - An Experience Sampling Study Creation Tool. In *BCS Health Informatics Scotland (HIS)* (Edinburgh, United Kingdom). BCS Electronic Workshops in Computing.

[64] Salganik, Matthew J. (2019). *Bit by Bit: Social Research in the Digital Age*. Princeton University Press,

[65] Silvia, Paul J, Kwapil, Thomas R, Eddington, Kari M and Brown, Leslie H. (2013). Missed Beeps and Missing Data: Dispositional and Situational Predictors of Nonresponse in Experience Sampling Research. *Social Science Computer Review* 31, 4. pp. 471-481.

[66] Stier, Sebastian, Breuer, Johannes, Siegers, Pascal and Thorson, Kjerstin. (2020). Integrating Survey Data and Digital Trace Data: Key Issues in Developing an Emerging Field. *Social Science Computer Review* 38, 5. pp. 503-516.

[67] Struminskaya, Bella, Lugtig, Peter, Keusch, Florian and Höhne, Jan Karem. (2020). Augmenting Surveys With Data From Sensors and Apps: Opportunities and Challenges. *Social Science Computer Review*. pp. 1-13.

[68] Tangmunarunkit, Hongsuda, Hsieh, Cheng-Kang, Longstaff, Brent, Nolen, S, Jenkins, John, Ketcham, Cameron,

Selsky, Joshua, Alquaddoomi, Faisal, George, Dony and Kang, Jinha. (2015). Ohmage: A General and Extensible End-to-End Participatory Sensing Platform. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6, 3. pp. Article 38, 01-21.

[69] Taylor, John T and Taylor, Wayne T. (2021). *Software Architecture: Foundations, Theory, and Practice.* Springer,

[70] Teamscope. *Teamscope: Data Collection App for Research | Secure, Offline.* https://www.teamscopeapp.com/, 2022. (Accessed 2/1/2023).

[71] Thai, Sabrina and Page-Gould, Elizabeth. (Dec 2018). ExperienceSampler: An Open-Source Scaffold for Building Smartphone Apps for Experience Sampling. *Psychological Methods* 23, 4. pp. 729-739.

[72] Urbandroid Team. *Don't Kill My App.* https://dontkillmyapp.com/, 2022. (Accessed 3/1/2023).

[73] Van Berkel, Niels, Ferreira, Denzil and Kostakos, Vassilis. (2017). The Experience Sampling Method on Mobile Devices. *ACM Computing Surveys (CSUR)* 50, 6. pp. Article 93, 01-40.

[74] Van Berkel, Niels, Ferreira, Denzil and Kostakos, Vassilis. (Dec. 2017). The Experience Sampling Method on Mobile Devices. *ACM Computing Surveys* 50, 6. pp. 93:1-93:40.

[75] W3C. *Geolocation API.* https://www.w3.org/TR/geolocation/, 2022. (Accessed 2/1/2023).

[76] Wenz, Alexander, Jäckle, Annette, Burton, Jonathan and Couper, Mick P. (2022). The Effects of Personalized Feedback on Participation and Reporting in Mobile App Data Collection. Social Science Computer Review 40, 1. pp. 165-178.

[77] Xiong, Haoyi, Huang, Yu, Barnes, Laura E and Gerber, Matthew S. (2016). Sensus: A Cross-Platform, General-Purpose System for Mobile Crowdsensing in Human-Subject Studies. In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (Heidelberg, Germany). Association for Computing Machinery. pp. 415-426.

A APPENDIX

A.1 Technical Overview of ESM Tools

Table 1: Overview of the technical aspects of the reviewed 30 tools that facilitate ESM study designs

Tool Name	Open Source	Programming Technology	Architecture: System	Architecture: Client	Data Exchange Format	Client Platforms
AWARE [20]	✓	PHP, MySQL, Vert.X, Flutter	client-server	native	JSON	Android, iOS
Enketo [14]	✓	JS, Node.js, Redis	client-server	web	JSON, XML	Android, iOS
ESMAC [5]	✓	Java	client-server	native	XML	Android
ESm Capture [15]		?	client-server	native	?	iOS
ExperienceSampler [71]	✓	JS, Apache Cordova	client-server	native (web)	JSON	Android, iOS
Formr [4]	✓	PHP, JS, R	client-server	web	JSON	Android, iOS
Funf [8]	✓	Python, Java, SQLite	client-server	native	JSON, binary	Android
iDialogPad [32]	✓	Objective-C	client-only, client-server	native	Custom	iOS
Jeeves [63]	✓	Java, Firebase, MongoDB	client-server	native	JSON	Android
LifeData [44]		?	client-server	native	?	Android, iOS
mEma [34]		?	client-server	native	?	Android, iOS
MetricWire [48]		?	client-server	native	?	Android, iOS
MobileQ [47]		?	client-server	native	?	Android
movisensXS [51]		?	client-server	native	?	Android
MyExperience [22]	✓	C#, .NET, SQL CE	client-only, client-server	native	XML	Windows Mobile
ODK [56]	✓	Java	client-server	native	XML	Android
ODK-X [55]	✓	Java, JS	client-server	native (web)	XML	Android
Ohmage [68]	✓	Java, PhoneGap, MySQL	client-server	native (web)	JSON, XML	Android, iOS
Paco [18]	✓	Java, Objective-C, AppEngine, GWT	client-server	native	JSON	Android, iOS
PartS [45]		?	client-server	native	JSON	Android
Piel Survey [9]		?	client-only	native	Custom	Android, iOS
Purple Robot [38]	✓	JS, Scheme, Java, Python	client-server	native (web)	JSON	Android
Qualtrics [61]		?	client-server	native, web	?	Android, iOS

Tool Name	Open Source	Programming Technology	Architecture: System	Architecture: Client	Data Exchange Format	Client Platforms
ResearchKit [26]	✓	Objective-C, Swift	client-only	native	-	iOS
Sema3 [40]		?	client-server	native	?	Android, iOS
Sensus [77]	✓	C#, Xamarin, Amazon AWS, R	client-server	native	JSON	Android, iOS
Survalytics [54]	✓	Java, Amazon DynamoDB	client-server	native	JSON	Android
SurveySignal [31]		?	client-server	web	?	Android, iOS
Teamscope [70]		?	client-server	native	?	Android, iOS
TEMPEST [6]		JS (?)	client-server	native (web)	JSON	Android, Web

A.2 Overview of ESM Tools for Study Administrators

Table 2: Overview of study-administrator-related aspects of 30 tools that facilitate ESM study designs

Tool Name	Probe Types: User	Probe Types: Sensor	Trigger Types	Study Authoring Facilities: GUI Editor	Preview	Live Stats	Additional Features & Concepts
AWARE [20]	Single choice, Multiple choice, Free text, Likert (scale), Quick answer, Photographic Effect Meter, URL Link, Date & time, Picture, Audio, Video	Location, Accelerometer, Magnetometer, Photometer, Barometer, Microphone, Camera, Battery, Bluetooth, Network, Application usage, Calls/ message, Email, Calendar, Gesture input	interv., signal, event	✓		✓	Focus on minimal battery impact, Concept of Sensors and Plugins (aka Services, also study specific ones), Plugins collect and analyse sensor data, Aware-Light Android application with passive sensor data collection

	Probe Types			Study Authoring Facilities			
Tool Name	User	Sensor	Trigger Types	GUI Editor	Preview	Live Stats	Additional Features & Concepts
Enketo [14]	Single choice, Multiple choice, Free text, Likert, Slider, Date & time, Ranking, Rating (Stars), Graphic/Map, Image, Table, Geo Location, Draw/Signature, Annotate Image, Audio, Video, Acknowledge/ Confirm	-	signal		?		Uses ODK XForms (external form builder tool available)
ESMAC [5]	Single choice, Multiple choice, Free text, Likert, Slider	Location, Ambient light, Bluetooth, Accelero-meter, Call log, Notifications, User activity, Weather, Display state	interv., signal, event	✓			Drag & drop question editor
ESm Capture [15]	Single choice, Multiple choice, Free text, Likert, Slider	Location	interv., signal, event	✓			Focus on ESM
Experienc eSampler [71]	Single choice, Multiple choice, Free text, Likert, Slider	-	interv., signal, event				Focus on ESM and non-technical users
Formr [4]	Single choice, Multiple choice, Free text, Likert, Slider, Date & time, Image choice, Colour, Rating, Geo location	-	interv., signal		✓		Concept of Runs (study protocols)

	Probe Types			Study Authoring Facilities			
Tool Name	User	Sensor	Trigger Types	GUI Editor	Preview	Live Stats	Additional Features & Concepts
Funf [8]	-	Location, Accelero-meter, Magneto-meter, Photometer, Barometer, Microphone, Camera, Battery, Bluetooth, Network, Gyroscope, Temperature, Pressure, Proximity, Gravity, Telephony, Browser bookmarks & searches, (Running) Applications, Screen, Rotation, Orientation, Contacts, Study Notifications, Call/SMS log	interv., signal, event				Dropbox integration, Web-based Android application generator (funf in a box), Concept of Pipeline (set of probes)
iDialogPad [32]	Single choice, Multiple choice, Free text, Likert, Slider, Affect Grid, Self-Assessment Manikin, Working-Memory Task, Image, Audio	Location	interv., signal, event		✓		Concept of a study as a computer program, Uses a proprietary script file (.qdf) for study configuration, Many pre-defined domain-specific question types, External study editor for .qdf files

Tool Name	Probe Types: User	Probe Types: Sensor	Trigger Types	Study Authoring Facilities: GUI Editor	Preview	Live Stats	Additional Features & Concepts
Jeeves [63]	Single choice, Multiple choice, Free text, Likert, Slider, Geo location, Date & time, Audio, Heart Rate, Image	Location, Accelero-meter, Ambient light, Gyroscope, Bluetooth, Microphone, Network	interv., signal, event	✓		✓	Block-based visual construction of studies, Custom patient 'attributes' (global state variables), Concept of End-User Development (EUD)
LifeData [44]	Single choice, Multiple choice, Free text, Likert, Slider, Date & time, Amount of time, Rating, Image/Photo, URL	Location	interv., signal, event	✓	✓	✓	Multi-language support, Delayed follow-up questions (via trigger), Yoking (linking dyads)
mEma [34]	Single choice, Multiple choice, Free text, Likert, Slider, Date & time, Image, Table, Geo location, Audio, Video	Location, Photometer, Microphone, Proximity, Motion, Humidity, Barometer, (claim to track all available sensors)	interv., signal, event	✓	?	✓	Question randomisation support, Participant group assignment, Dynamic baseline scoring & branching, GARMIN device support
MetricWire [48]	Single choice, Multiple choice, Free text, Likert, Slider, Date & time, Geo location, Percentage, Percentage fill, Ranking, Image, Audio, Video	Location, Geofencing	interv., signal, event	✓	?	✓	Focus on ESM, Study protocols can adapt to previous individual behaviours
MobileQ [47]	Single choice, Multiple choice, Free text, Likert, Slider, Grid	-	interv., signal	✓	✓	✓	Roles: project manager, field operator, Surveys configured in tree-structure, Drag & drop question editor

	Probe Types			Study Authoring Facilities			
Tool Name	User	Sensor	Trigger Types	GUI Editor	Preview	Live Stats	Additional Features & Concepts
movisens XS [51]	Single choice, Multiple choice, Free text, Likert, Slider, Date & time, Geo location	(still in beta) Location, Accelerometer, Network (traffic), Battery, Display, Phone/SMS (meta-data), Bluetooth	interv., signal, event	✓	?	✓	Specifically targets ESM research, Extensive documentation, List of supported devices, Multi-language support, Provides custom devices for ESM studies
MyExperience [22]	Single choice, Multiple choice, Free text, Video, Audio, Photo	Location, Bluetooth, Network, Microphone, Phone calls, Application usage, Calendar	interv., signal, event		?		Concept of Sensors, Triggers, and Actions, Sensor state changes are logged; can be used by triggers Sensors can conditionally be started and stopped
ODK [56]	Single choice, Multiple choice, Free text, Likert, Slider, Date & time, Audio, Images, Binary files	-	interv., signal				Toolkit
ODK-X [55]	Single choice, Multiple choice, Free text, Likert, Slider, Date & time, Audio, Images, Binary files	Location	interv., signal				Toolkit, Detailed error analytics
Ohmage [68]	Single choice, Multiple choice, Free text, Audio, Video, Image, Binary file	Location, Accelerometer, Network, Microphone, Camera, Call/message logs, System usage	interv., signal, event	✓		✓	Concept of Campaign, Survey and Prompt, Concept of Class and Participants, Modular tool architecture (separate tools for specific use cases), Built for (passive) participatory data sensing
Paco [18]	Single choice, Multiple choice, Free text, Likert, Geo location, Rating (smiley), Audio, Photo	Device details (Android), Application usage, User present, Phone on/off	interv., signal, event	✓	✓	✓	Easy Google sign-in process, Concept of Experiments (studies), Concept of Triggers and Actions, Concept of Inputs and Surveys, Concept of Data Collectors (survey & sensor), Custom scripting support (JavaScript)

Tool Name	Probe Types: User	Probe Types: Sensor	Trigger Types	Study Authoring Facilities: GUI Editor	Preview	Live Stats	Additional Features & Concepts
PartS [45]	Single choice, Multiple choice, Free text	Location, Network, Bluetooth, Screen, Battery, Device details, Running applications	interv., signal, event	✓		✓	News section (published to all participants), Participant mail box, Collaboration features (comments and linking of visualisations), Discussion boards, In-app participant-administrator chat, Allows combined events for triggers
Piel Survey [9]	Single choice, Multiple choice, Free text, Slider	-	interv., signal		✓		Uses proprietary text file (Control File) for study protocol configuration
Purple Robot [38]	-	Accelerometer, Battery, Bluetooth, Gyroscope, Light, Magnetometer, Pressure, Proximity, Network, Call state, Application metrics, Running applications, Screen, Device details	interv., signal, event				
Qualtrics [61]	Single choice, Multiple choice, Free text, Likert, Slider, Ranking order, File Upload, Side by Side, Constant Sum, Hot Spot, Graphic Slider, Pick-Group-and-Rank, Heat Map, Drill Down, Highlight/Mark, Signature, Captcha Verification, Meta-Info Question, Screen Capture, Timing	-	interv., signal	✓	✓	✓	Survey validation/review features (with 'expert' help), Multi-language support, Drag & drop question editor, Question & choice randomisation, Survey preview mode (mobile view), Comprehensive question library, Theming support/library, Automatic reports

Tool Name	Probe Types: User	Probe Types: Sensor	Trigger Types	Study Authoring Facilities: GUI Editor	Preview	Live Stats	Additional Features & Concepts
ResearchKit [26]	Single choice, Multiple choice, Free text, Likert, Slider, Date & time, Geo location, Email	Location, Accelerometer, Gyroscope, Microphone, Multi-Touch display	interv., signal, event			✓	Toolkit Concept of Surveys, Consents, and Tasks, Multi-language support, HealthKit integration, Many pre-defined medical Active Tasks (Stroop, range of motion, tapping speed, etc.)
Sema3 [40]	Single choice, Multiple choice, Free text, Slider, Grid, Combination-choice slider	- (?)	interv., signal	✓	✓	✓	Visualisations can be shared with participants
Sensus [77]	Single choice, Multiple choice, Free text, Slider, Audio	Location, Accelerometer, Compass, Network, Microphone, Barometer, Battery, Photometer, Call/ messages	interv., signal, event	✓	✓		Focus on non-technical users, Concept of Mobile Crowd Sensing (MCS), Combines survey editor and runtime in one application, Concept of Sensing Plans, Studies are distributed via email or URL, Support for 'RCTs' (random assignment of protocols upon start)
Survalytics [54]	Single choice, Multiple choice, Free text, Slider	Location, Application metrics	interv., signal, event				Forms can be created in Google Sheets (via custom script)
SurveySignal [31]	(uses external questionnaires)	-	interv., signal		?	✓	Signals via SMS or WhatsApp messages with link to survey, Compatibility screening test option, Participant randomisation support (RCT)
Teamscope [70]	Single choice, Multiple choice, Free text, Date & time, Image, Geo location	Location	interv., signal, event	✓	?	✓	Query manager for resolving data inconsistencies, Team management (groups, rights & roles)
TEMPEST [6]	Single choice, Multiple choice, Free text, Slider	Location	interv., signal, event	✓	✓		Study protocol modelled as a computer program, Uses concept of widgets & screens, Widgets/triggers can be used for calculations, End-user programming

A.3 Overview of ESM Tools for Study Participants

Table 3: Overview of participant-related aspects of 30 tools that facilitate ESM study designs

	Usability & UX		Privacy	
Tool Name	Special Features	Offline Mode	Special Features	Certification
AWARE [20]	Invitation via QR-code or link	✓	User control of active sensors, One-way hashing of personal identifiers	
Enketo [14]		✓		
ESMAC [5]		✓		
ESm Capture [15]		✓		
ExperienceSampler [71]	Snooze button (delay answering of questionnaires)	✓		
Formr [4]			Unlinking of surveys (random order results), Hiding of results, No storage of user ids or dates, Minimum number of answers requirement,	
Funf [8]		✓	Stop data collection button, All sensor data stored on phone, Export must be manually triggered by participant, All data locally encrypted	
iDialogPad [32]	Sleep button to suspend all surveys	✓		
Jeeves [63]	Chat messaging systems with study administrators	✓		
LifeData [44]	See own response rates (monitoring)	✓	Anonymous mode, HIPAA & GDPR compliant	✓
mEma [34]	Participants can initiate sampling events	✓	Audio recordings stay on local device and get deleted after being processed, HIPAA compliant	✓
MetricWire [48]		✓		✓
MobileQ [47]	Demo mode, Customise start & end times, Customise notification modality (vibration and sound)	✓		
movisensXS [51]		✓	Server hosted in EU (Germany), ISO 27001 certificate, All data collected via pseudonyms	✓
MyExperience [22]		✓		

Tool Name	Usability & UX Special Features	Offline Mode	Privacy Special Features	Certification
ODK [56]		✓		
ODK-X [55]		✓		
Ohmage [68]	Comparison of own data with baseline, Self-registration support for public studies	✓		
Paco [18]		✓		
PartS [45]	Participant chat and mailbox with administrators, News section (published to all participants), In-app overview of available studies, In-app overview of all on-device captured data	✓	Suspend data capturing button, In-app overview of all accessed sensors before enrolment, Exclusion of situational-specific services (sensor readings) by participant	
Piel Survey [9]		✓	GDPR compliant, Complete control over data transfer process	✓
Purple Robot [38]		✓		
Qualtrics [61]		✓	ISO 27001 certification	✓
ResearchKit [26]		✓		
Sema3 [40]	Visualisation of own responses	✓		
Sensus [77]	Invitation via email or link	✓	In-app overview of all accessed sensors before enrolment, Anonymisation control: latitude, longitude, device id, timestamp (set by administrator), Field omission, hashing, random-anchoring of time, rounding/truncating, Customisable data polling rate (set by administrator)	
Survalytics [54]		✓	Back end data store (Amazon) is HIPAA & PII compliant	
SurveySignal [31]	Invitation via SMS or WhatsApp link			
Teamscope [70]		✓	HIPAA, GDPR, GCP compliant	✓
TEMPEST [6]		✓		

Received October 2022; revised Jnauary 2023; Accepted April 2023.