Development, Deployment, and Evaluation of DyMand: Wearable AI for In-Situ Detection and Capture of Couples' Dyadic Interactions in Chronic Disease Management

GEORGE BOATENG, ETH Zurich, Switzerland
PRABHAKARAN SANTHANAM, ETH Zurich, Switzerland
ELGAR FLEISCH, ETH Zurich, Switzerland and University of St. Gallen, Switzerland
JANINA LÜSCHER, Swiss Paraplegic Research, Switzerland
THERESA PAULY, Simon Fraser University, Canada
URTE SCHOLZ, University of Zürich, Switzerland
GUY BODENMANN, University of Zurich, Switzerland
TOBIAS KOWATSCH, ETH Zurich, Switzerland, University of Zurich, Switzerland, and University of St. Gallen, Switzerland

Dyadic interactions of couples are of interest as they provide insight into relationship quality and chronic disease management. Existing sensing systems primarily infer social structure within groups retrospectively, or capture couples' interactions at random or scheduled times, which could miss meaningful interpersonal interactions between partners. In this work, we developed a smartwatch-based algorithm that detects interaction moments between partners and enables interaction-aware data capture in everyday settings. It uses the Bluetooth signal strength between two smartwatches, each worn by one partner, and a voice activity detection machine-learning model that we trained, to infer that the partners are interacting and then trigger data collection. We operationalized this sensing approach by developing, deploying, and evaluating DyMand, an open-source smartwatch and smartphone system that integrates multimodal sensor and self-report data collection in situ. We deployed the DyMand system in a 7-day field study and collected and analyzed data about social support and diabetes discussions from 13 (N=26) Swiss-based heterosexual couples managing diabetes mellitus type 2 of one partner. Our system evaluation showed that 77.6% of algorithm-triggered recordings contained partners' conversation moments compared to 43.8% for scheduled triggers, and that DyMand was easy to use. Preliminary insights into disease management behavior revealed that 5.2% of couples' conversation moments contained social support, and partners discussed diabetes management 2.4% of the time based on the audio data analyses. In contrast, the self-report data showed that patients received support in 67.7% of couples' interactions and caregivers provided support in 71.8% of couples' interactions. DyMand enables insights into dyadic behavior in-situ, and could be used by social, clinical, or health psychology researchers to understand the social dynamics of couples that are managing chronic diseases in everyday life to inform the development and delivery of behavioral interventions.

Authors' addresses: George Boateng, gboateng@ethz.ch, ETH Zurich, Zurich, Switzerland; Prabhakaran Santhanam, psanthanam@ethz.ch, ETH Zurich, Zurich, Switzerland; Elgar Fleisch, efleisch@ethz.ch, ETH Zurich, Zurich, Switzerland and University of St. Gallen, St. Gallen, Switzerland; Janina Lüscher, janina.luescher@paraplegie.ch, Swiss Paraplegic Research, Nottwil, Switzerland; Theresa Pauly, theresa_pauly@sfu.ca, Simon Fraser University, Vancouver, Canada; Urte Scholz, urte.scholz@psychologie.uzh.ch, University of Zürich, Zurich, Switzerland; Guy Bodenmann, guy.bodenmann@psychologie.uzh.ch, University of Zurich, Zurich, Switzerland; Tobias Kowatsch, tkowatsch@ethz.ch, ETH Zurich, Zurich, Switzerland and University of Zurich, Zurich, Switzerland and University of St. Gallen, St. Gallen, Switzerland.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM 2637-8051/2026/7-ART
https://doi.org/10.1145/3833380

Just Accepted

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing**; • **Applied computing →** *Consumer health*; *Psychology*.

Additional Key Words and Phrases: Multimodal Sensor Data; Couples; Smartwatches; Smartphones; Mobile Computing; Wearable Computing; Machine Learning; Speech Processing; Chronic Disease Management; Social Support

1 Introduction

Romantic relationships have powerful effects on people's mental and physical health (see [71] for an overview). For instance, conflicts and negative qualities of one's intimate relationship are associated prospectively with morbidity and mortality [57]. Romantic or social relationships play an important role in illness management if partners share the responsibility and consider the disease to be their joint problem instead of being only the problem of the afflicted partner [68, 75] and it can involve social support and common dyadic coping (CDC) [17]. Social support entails providing resources to help a receiver cope in a time of need and can be emotional (e.g., providing comfort or encouragement) or instrumental (e.g., help with practical problems and tasks; [32, 59, 74]). CDC is the "we approach" to dealing with stressors in a couple's relationship [27] which can be assessed objectively by counting first-person plural pronouns [72], by questionnaire or behavioral observation. Social support among couples and CDC in chronic disease management have been shown to have mostly positive effects on emotional well-being [28, 47, 73, 82], and result in healthier eating habits among diabetes patients [63]. Consequently, it is of interest to better understand couples' dyadic interactions in-situ, for example, in couples' management of diabetes in daily life [50, 58] as they could inform the development and delivery of behavioral interventions to, for example, improve emotional well-being, physical activity, diet, and medication adherence.

While prior work has demonstrated the importance of dyadic processes such as social support and common dyadic coping in chronic disease management, capturing these processes in situ remains challenging. Existing ambulatory approaches typically rely on random or scheduled sampling (e.g., periodic audio snippets or self-reports), which are agnostic to whether meaningful interaction is actually occurring (see Section 2). As a result, these methods often miss precisely those moments when dyadic processes unfold—namely, when partners are co-present and actively engaging with one another.

Ubiquitous devices such as smartphones and smartwatches provide a good opportunity to collect relevant data such as sensor and self-report data from couples in daily life. Smartwatches in particular could be leveraged to collect data on couples' dyadic interactions and chronic disease management. Several features of smartwatches make them uniquely positioned for this task. Firstly, they are mostly with the wearer since they are worn on the wrist in comparison with a smartphone which could be in various places like the pocket, or bag, and just not in proximity to the user, or devices like Amazon Echo or Google Home which can only be in one place and not always around the owners. Additionally, commercial smartwatches could be used to collect a wide variety of sensor data such as audio (for analysis of social support and disease discussions), Bluetooth (for proximity detection), heart rate, accelerometer, and gyroscope (for physical activity), and ambient light (to detect context). Finally and importantly, smartwatches could be leveraged to detect and capture dyadic interactions of partners (e.g., triggering data collection when partners are close and speaking) as we do in this work.

In this work, we developed a novel smartwatch-based algorithm that uses the Bluetooth Low Energy (BLE) signal strength between two smartwatches, each worn by one partner, and a voice activity detection (VAD) machine-learning model that we trained, to infer in real-time that the partners are interacting, and then trigger data collection. We operationalized this sensing approach by developing, deploying, and evaluating DyMand, an open-source software system that uses a smartwatch [1], and smartphone [2] for ambulatory assessment of couples' **Dy**adic **Man**agement of chronic **d**iseases in daily life. The DyMand system is designed around this sensing

[1] https://bitbucket.org/mobilecoach/dymandwatchclient/src/master/
[2] https://bitbucket.org/mobilecoach/dymand-mobilecoach-client/src/master/

Just Accepted

approach: detecting dyadic interaction moments in real time using lightweight, on-device signals, and leveraging those moments to trigger contextually relevant capture of self-report and sensor data. Consequently, the system enables targeted and minimally intrusive sampling of behavior as it unfolds in naturalistic settings. We deployed the system in a field study with 13 heterosexual cohabiting couples in Switzerland that are managing type 2 diabetes (T2DM) of one partner, a common chronic disease affecting 6% of the Swiss population [46]. We collected multimodal sensor data and self-report data about partners' social support and diabetes management in daily life. We evaluated the technical performance, usability, and utility of DyMand. In particular, to demonstrate the utility of the DyMand system in capturing partners' conversation moments, we analyzed the audio interaction data to understand (1) how often partners provided social support and the percentage of the two types of social support — instrumental and emotional social support in the management of T2DM and (2) how often both partners discussed diabetes or diabetes-related issues. Furthermore, we analyzed the self-report data to understand how frequently patients and caregivers received and provided social support, respectively, and their perception of how closely they were managing T2DM together. This kind of insight enabled by DyMand has the potential to provide a sound basis for theory- and evidence-based development of dyadic interventions in the context of couples' dyadic illness management. DyMand could be used by social, clinical, or health psychology researchers to understand the social dynamics of couples that are managing chronic diseases in everyday life to inform the development and delivery of behavioral interventions. This work builds upon our study protocol published in 2019 [58] and is an extension of our prior preliminary work [19, 24]. This work provides a more detailed description of the DyMand system and its real-world deployment and evaluation.

1.1 Contribution

Our primary contribution is a novel real-time dyadic sensing algorithm that uses smartwatch-based BLE proximity + on-device VAD to detect interaction moments between dyads in naturalistic settings. Our secondary contributions are as follows:

(1) A deployable open-source software system — DyMand — consisting of a smartwatch, smartphone, and server, that operationalizes the dyadic sensing approach and integrates in-situ multimodal sensor and self-report data collection.
(2) A 1-week field deployment and evaluation of DyMand with 13 couples managing T2DM, demonstrating feasibility in adequately capturing couples' interaction moments in chronic disease contexts.
(3) Initial empirical insights into social support, CDC and disease discussions in couples' chronic disease management based on collected sensor and self-report data.

This paper primarily contributes a novel smartwatch-based algorithm for detecting dyadic interactions in situ. The DyMand system and field deployment serve to instantiate and evaluate this approach, demonstrating its feasibility and illustrating the kinds of insights such an approach can enable.

The rest of this paper is organized as follows. Next, we discuss related work (Section 2). We then describe the system design (Section 3) and its implementation (Section 4). In Section 5, we describe the deployment of the system in a user study, and, in Section 6, we evaluate its technical performance and usability. In Section 7, we provide self-report and audio-derived insights into various constructs in the context of chronic disease management. Finally, we describe limitations and future work in Section 8 and conclude this work in Section 9 with a summary.

2 Related Work

In this section, we describe related works that have developed approaches for (1) sensing social interactions, and (2) ambulatory data collection within the specific context of couples' interactions in daily life.

Just Accepted

2.1 Sensing Social Interaction

Several works leveraged various signals such as Bluetooth, infrared (IR) sensors, Radio Frequency Identification (RFID), accelerometer, and audio for inferring social interaction among groups of individuals and dyads mostly done by the research group of Alex Pentland at the MIT Media Lab [15, 30, 31, 33, 34, 39, 40, 45, 60, 64, 66, 76]. Several of these works have used sensors in mobile devices. In one of the pioneering works in the field of social sensing in the Reality Mining project, Eagle and Pentland inferred social partners, organizational rhythm, and relationship types (e.g., friends, colleagues) among 100 individuals using data from Nokia 6600 devices such as Bluetooth (for proximity), time and location data collected over 9 months [33]. A follow-up study by Eagle et al. used Bluetooth sensors in mobile phones to detect physical proximity and infer friendship networks [34]. Sekara and Lehmann used Bluetooth signal strength in mobile devices to infer physical proximity, and along with other signals such as WiFi, locations, etc, inferred the strength of friendships in a study with 134 students [76]. Banerjee et al. developed the Microsoft Virtual Compass which is a smartphone-based system that uses WiFi and Bluetooth signals in mobile phones to infer physical proximity and then social interactions between individuals [15].

Several other works have used custom hardware like wearables for inferring social interactions. In another one of the pioneering works, Choudhury and Pentland developed the Sociometer, a wearable device that has a microphone, an accelerometer, and an IR sensor that was used to detect proximity between subjects, and which two individuals were engaged in a conversation [31]. Olguin et al. developed an updated version called the Sociometric badge [3] which additionally included a Bluetooth module and implemented real-time inference of physical activity, extraction of speech features in real-time for offline analysis to determine when an individual spoke and face-to-face interaction detection using the IR sensor when two people wearing badges are facing each other within a 30-degree cone and 1-m distance [64]. Similarly, the SocioPatterns platform [4] is a wearable that uses RFID sensors that emit signals within 1 to 1.5 meters for collecting proximity data to understand the dynamics of individuals in social interactions [30]. Rabbi et al. developed a wearable system consisting of several sensors — a 3-axis accelerometer, barometer, light sensor, temperature sensor, humidity sensor, compass, and microphone — for inferring the mental well-being of individuals in daily life [66]. One component of the system inferred which two or more people engaged in a conversation by computing the mutual information of speech features extracted from the microphones for different pairs of individuals. Matic et al. detected face-to-face social interaction on a smaller spatiotemporal scale (4 - 6 people) using the compass in mobile phones to infer orientation, WiFi for proximity, and a chest-worn accelerometer to infer speech activity [60]. Hung et al. used a sensor pack worn around the neck consisting of a 3-axis accelerometer, a proximity sensor, and an indoor positioning system to collect data from 32 individuals interacting at a social event and used only accelerometer data to infer conversation groups from the data [45]. A follow-up work by Gedik and Hung similarly used a 3-axis accelerometer for inferring conversation groups [39]. Girolami et al. developed SocializeME, a framework consisting of an Android app that communicates with wearable BLE tags and a remote server to monitor and detect social interactions among individuals [40].

Despite these advances, existing approaches exhibit three key limitations for studying dyadic interaction in everyday life. First, many of these systems infer interactions retrospectively from group-level data rather than detecting them in real time (e.g., [31, 33, 34, 64, 66]). Given that computations need to be done for all pairs within the group in a study, these works do not perform real-time detection of social interaction on-device but rather perform retrospective analysis offline to infer social interaction dynamics. Second, proximity is often used as a proxy for interaction, leading to false positives when individuals are co-located but not engaged [15, 30, 39, 40, 45]. These works assume physical proximity alone is a good proxy for face-to-face interaction, which could include

[3] https://hd.media.mit.edu/badges/
[4] http://www.sociopatterns.org/

Just Accepted

several false positives, such as two people just sitting close in, say, a bus, office, or classroom but not interacting Third, these systems are not designed for fixed dyads with known relationships, where the goal is not to infer social structure but to capture interaction dynamics as they occur. They are focused on modeling human behavior and interaction dynamics on a large scale based on data from several individuals (for example, given this large group of individuals, which of them are friends vs colleagues [33, 34], or which conversational groups they belong to [39, 45]). However, they do not detect social interactions at a much smaller spatiotemporal scale such as detecting conversation moments between a specific romantic couple dyad. These limitations motivate the need for a sensing approach that combines proximity with interaction cues and operates in real time at the dyadic level.

Our work builds upon several of the ideas and approaches in these prior works such as using BLE to infer proximity between individuals and speech for detecting conversation and contributes to the specific context of couple dyads managing a chronic disease in daily life. In particular, unlike prior works that retrospectively infer relationship type, conversation groups, and social interaction between two or more individuals within a group using signals, our focus is **detecting conversation moments in-situ** in **real-time on-device within fixed dyads** that we know a priori to have a relationship (romantic couples). Hence, our task is as follows: given this specific pair of individuals, detect in real-time all moments that they have interactions with each other. This task poses unique challenges and differs from the tasks in prior work as follows: given collected data from N number of people, infer all pairs or subgroups that had some social interaction or have a specific relationship type.

We implemented a lightweight system that runs on-device on a smartwatch and performs real-time detection of partners' interaction moments using not only BLE for proximity like most prior work (which erroneously assumes that proximity implies interaction) but additionally, a VAD algorithm to then trigger the collection of sensor and self-report data that are relevant for understanding couples' dyadic management of chronic diseases in daily life. Unlike several of the systems such as [31, 40, 60], our system does not need additional hardware which could be burdensome to carry but uses a consumer device — a smartwatch. Our use of a smartwatch increases the likelihood of capturing interaction moments as they are mostly with the wearer, compared to a mobile phone or a custom device that has to be additionally worn. Also, our real-time detection on a smartwatch provides opportunities for just-in-time interventions and smartwatch-based dyadic mood-centric interactions such as in those in [56] and yet poses unique challenges for algorithm development and implementation which we address in this work. Furthermore, we deployed our system in the context of chronic disease management, in which we collected and evaluated relevant constructs such as frequency and type of social support provided, and frequency of diabetes discussion in partners' diabetes management in daily life.

2.2 Ambulatory Data Collection among Couples

Several mobile sensing platforms such as mCerebram [43] and AWARE [36] have been developed for collecting sensor and self-report data from individuals in daily life. Various smartphone applications have been developed for ambulatory data collection in the context of couples' interactions in daily life. For example, the Electronic Activated Recorder (EAR) [61, 69], a smartphone-based application has been used in several studies, especially for the collection of audio data in various couples' interactions such as couples managing breast cancer [49, 70]. The EAR triggers data collection at random or scheduled times in the day and collects snippets of ambient sound (e.g., 50 seconds every 9 minutes), which are later transcribed and coded. The EAR does not collect self-report data. On the other hand, mobile and wearable systems were used to collect sensor and self-report data for conflict detection [78, 79] and stressful conversations detection [16] among couples. Similar to the EAR, audio data were collected at random or scheduled times in the day (3 minutes of audio every 12 minutes). Other works perform all-day audio/video recording such as work [67] that used a digital recorder for a whole-day recording of couples managing cancer in daily life and [80] that also performed all-day video recording of couples.

Just Accepted

Despite these advances in the ambulatory assessment of couples' interactions, there are still gaps. Firstly, any random or scheduled triggering of data collection does not take advantage of the dyadic nature of couples' interactions (e.g., by inferring if partners are interacting) and could miss key conversations/interaction moments. Secondly, the EAR collects only audio and does not leverage potentially more information from other sensor data or self-reports. Finally, systems that perform all-day recordings of audio or video (e.g., [67, 80]) pose significant privacy concerns. While these approaches have yielded valuable insights, their reliance on random or scheduled sampling introduces a fundamental mismatch between data collection and the phenomenon of interest. Dyadic processes such as social support and CDC are episodic and context-dependent; sampling that is decoupled from interaction moments risks both missing relevant events and capturing large amounts of irrelevant data. This limitation suggests the need for interaction-aware sampling, in which data collection is triggered by the detection of meaningful interpersonal engagement.

2.3 Research Gap

Fundamentally, prior sensing systems have largely focused either on (a) inferring social structure within groups retrospectively, or (b) capturing behavior independent of dyadic interaction context, i.e., randomly or at scheduled times. No system exists that detects interaction moments within known dyads in real time and uses those moments to trigger targeted data collection. This gap limits our ability to study how interpersonal processes unfold dynamically in everyday life, particularly in contexts such as chronic illness management, where timing and context are critical. Our system fills the current gap by implementing a novel algorithm that detects partners' conversation moments using consumer smartwatches and then triggers the collection of sensor and self-report data such as social support between partners and frequency of discussing diabetes management, which is relevant for better understanding couples' dyadic management of diabetes in daily life. To the best of our knowledge, this work is the first that implements a real-time smartwatch-based system that uses BLE and VAD to detect interaction moments of romantic partners in daily life and collects and evaluates relevant constructs in couples' dyadic management of chronic diseases. Unlike prior work, DyMand uniquely combines real-time dyadic detection with multimodal capture in naturalistic chronic illness contexts. DyMand contributes a distinct approach that shifts from retrospective inference and schedule-based sampling toward real-time, interaction-aware sensing in fixed dyads. By combining proximity estimation with voice activity detection directly on wearable devices, our approach reduces false positives associated with proximity-only methods and enables targeted data capture during likely interaction moments. Consequently, DyMand enables studying interpersonal processes as they unfold in situ, with implications for both behavioral research and intervention design.

3 Development: System Design

In developing DyMand, experts from the field of computer science, information systems, and health psychology used justificatory knowledge from prior work [22, 47, 58, 63, 79] about social support, CDC, health behavior, and emotional well-being to derive a list of design specifications that are important for collecting corresponding data in-situ, in the context of chronic disease management. We describe the specifications.

3.1 Physical Closeness Monitoring

The system should track the physical closeness of the partners during waking hours. This information can be used to infer how much time romantic partners spend in various forms of daily interactions (engaging in a shared activity, talking, and arguing), which could be used to predict the couple's relationship outcomes [41]. Furthermore, physical closeness tracking can be used in combination with other kinds of data to capture moments when partners are interacting, which would enable the collection of data relevant to chronic disease management such as emotional well-being, social support, and CDC based on partners' interaction context.

Just Accepted

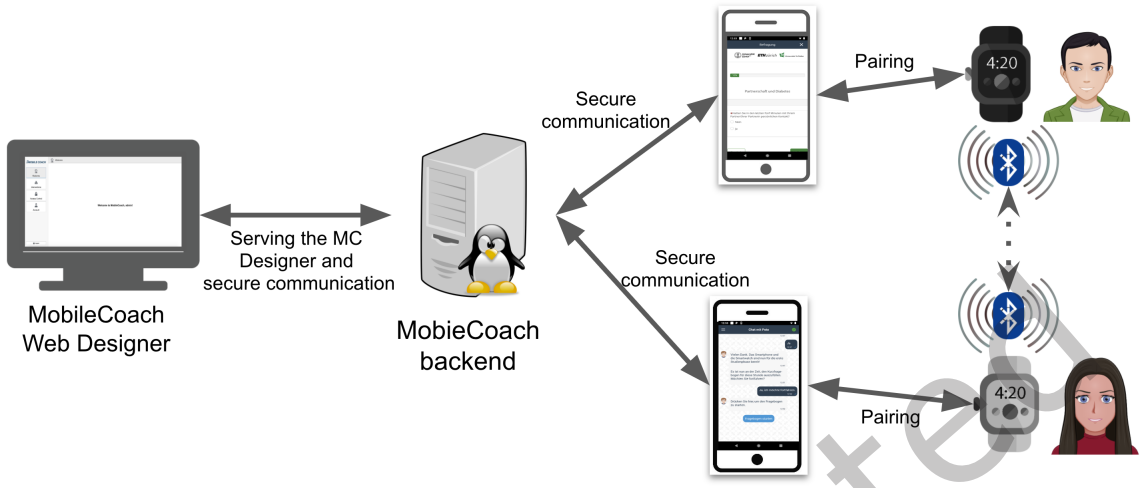


Fig. 1. Overview of the DyMand system

Just Accepted

3.2 Multimodal Sensor Data Collection

The system should collect relevant 5 minutes worth of multimodal sensor data (in particular audio) once each hour during waking hours (set by couples), ideally when partners are interacting. We restricted the data collection requirement so that we collect only one 5-minute sample per hour (audio data) for privacy reasons to ensure that we do not collect too much audio of the couples' daily life. Furthermore, the requirement for collecting this data when partners are interacting ensures there is a high likelihood of capturing conversations between partners as compared to collecting data at a random or scheduled time. Sensor data such as audio, heart rate, gestures, and physical activity could be used to manually and automatically infer behavioral information such as social support, CDC, emotional well-being, and health behavior, which are relevant for chronic disease management [58]. Audio, for example, can be used to code and analyze constructs such as the frequency of social support and frequency of discussion about diabetes management which we do (see Section 6). Also, audio together with other data such as heart rate and movement data could be used to automatically detect the emotions of each partner [13].

3.3 Self-report Data Collection

The system should collect self-report data immediately after sensor data collection and at the end of the day. This requirement ensures that various validated self-reported instruments also collect relevant data about social support, CDC, emotional well-being, and health behavior. Collecting this data right after sensor data collection enables matching the responses to the inferences made from the sensor data based on the partners' interactions. Additionally, collecting this data at the end of the day enables relevant daily summative data such as whether the patient took their medication for the day.

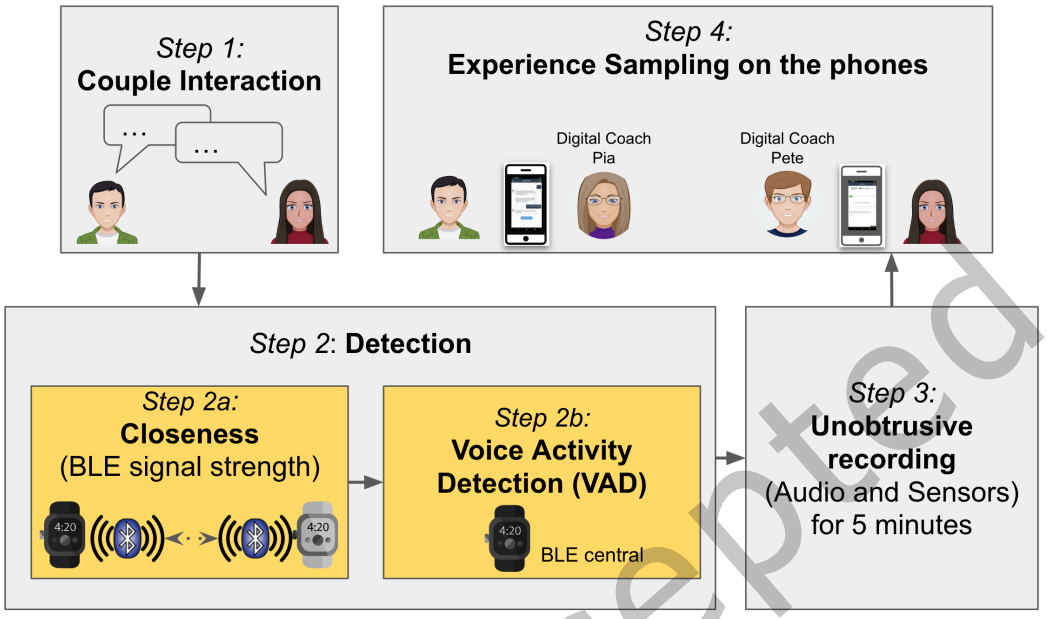


Fig. 2. Use case of DyMand system

4 Development: System Implementation

In this section, we describe our implementation of the system requirements to develop the DyMand system consisting of an overview of the DyMand system, devices used, the MobileCoach platform, a smartphone app, and a smartwatch app.

4.1 Overview of the DyMand system

The DyMand system (Figure 1) consists of a smartwatch app (section 4.5), and a smartphone app (section 4.4) built on top of the MobileCoach platform (section 4.3) [37, 53] that consists of a web-based intervention designer and backend. Each partner is given a smartwatch and a smartphone (both paired) running the DyMand apps.

The smartwatch app collects 5-mins of sensor data (section 4.5.4) based on the detection of the partners' interactions via Bluetooth Low Energy (BLE) and Voice Activity Detection (VAD) (Figure 2). After data collection, it gives a vibration alert on the smartwatch, and then sends an intent to the smartphone app to also give an alert (push notification) and trigger the self-report for each partner to complete separately. The smartphone app then sends a signal to the MobileCoach backend (Appendix B) to trigger the showing of the self-report on the smartphone. The smartphone app is customized with two digital coaches PIA (interacting with the partner with diabetes) and PETE (interacting with the partner without diabetes) that send various messages to each partner (e.g., "it is time to complete the self-report"). MobileCoach is also used for collecting data from the partners during the setup phase of the devices and also triggers end-of-the-day diary questionnaires. MobileCoach also sends reminders and escalation messages when the expected number of self-reports is not completed.

Just Accepted

4.2 Devices: Polar M600 Smartwatch and Nokia 6.1 Smartphone

For implementing the data collection from partners during their conversation moments, we decided to use two smartwatches (one for each partner) as they are more likely to be with wearers often compared to a smartphone, and could better capture physical closeness and interactions. Furthermore, a smartwatch has a better chance of capturing conversations if audio is recorded on it than a smartphone that may not be in the proximity of the partners. Additionally, relevant data of interest such as heart rate and gestures can be captured on a smartwatch and not a smartphone.

We started working with Apple's WatchOS platform but to the best of our knowledge at the time of development, we could not find a solution to start an audio recording as a background process. Hence we moved on to use smartwatches with the Wear OS (based on Android) platform because it provided a lot of flexibility for data collection. We chose a device from the company, Polar as their devices are regularly used in research [11] and we found the Polar M600 smartwatch to be practically priced and its battery seemed to last well with continuous sensing and with a daily charge cycle. Additionally, the heart rate sensor had good performance in a comparison study in which it was shown to be accurate during periods of steady-state activities like cycling, walking, jogging, and running, but less accurate during some exercise intensity changes [42].

We decided to use a smartphone to collect self-report data as the screen size of the watch was too small for filling out self-reports, and to serve as an intermediary between the MobileCoach server and the smartwatch app. We used two Nokia 6.1 smartphones running Android 9.0 — one per partner — which we outfitted with SIM cards for Internet access. We found Nokia 6.1 to be practically priced, with a moderate screen size (5.5") and with good haptics.

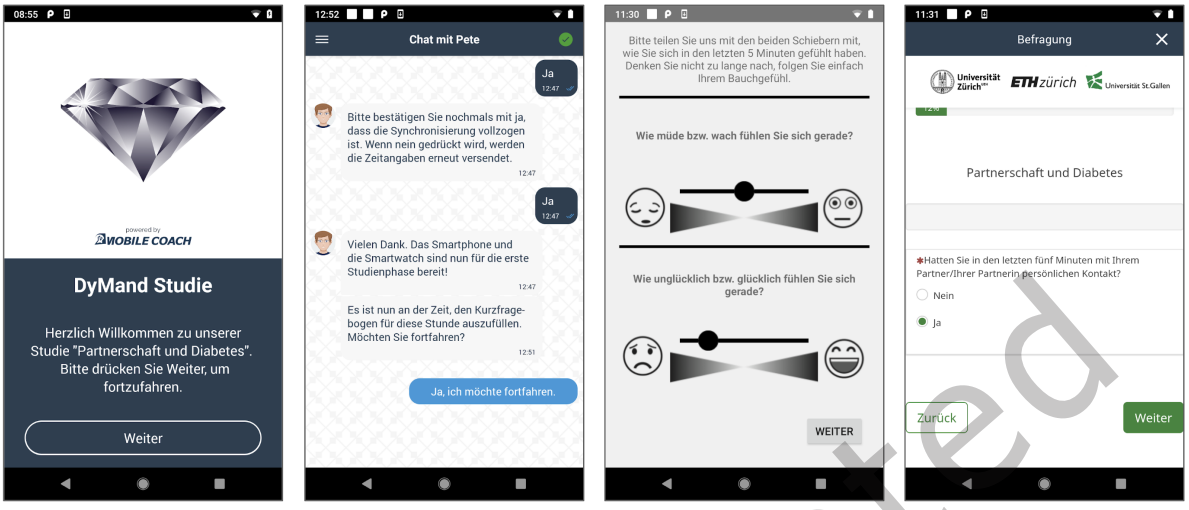
4.3 MobileCoach Platform

We implemented the DyMand system on top of MobileCoach, an open-source software platform for the design of behavioral interventions and ecological momentary assessments [37, 53]. MobileCoach provides a server setup and a mobile client app. The server with its backend setup (see Appendix A) offers a web-based intervention designer, called MobileCoach Designer (see Appendix B). The intervention author can use this interface to design dialogues that can be sent to the mobile app. The author can also put together rules in the designer and connect them to these dialogues so that the messages can be sent when the conditions in these rules are satisfied. The rules could be based on intent from the user (e.g., button press in the mobile app), time range (e.g., between 2 and 3 pm), and other variables which can be created and used in the designer. When the mobile app is opened for the first time, it can show information to the user about the app and let the user pick a conversational agent, i.e. a computer program that imitates the conversation with a human being [52]. By default, the app provides two conversational agents and after one is chosen, the agent can communicate with the user. The agent converses using the dialogues designed in the MobileCoach Designer. With some customizations and a smartwatch app connected with the MobileCoach framework, it fitted the DyMand system's use case.

4.4 MobileCoach Smartphone App

MobileCoach provides a skeleton app in React Native, which is a framework for building cross-platform applications for Android and iOS devices. We customized it for the DyMand system and used only the Android app. We customized it to have the two conversational agents PIA (interacting with the partner with diabetes) and PETE (interacting with the partner without diabetes). The smartphone app acts as an intermediary between the smartwatch and the server. It relays the "user intent" messages that indicate that "a recording in the smartwatch has been completed" to the server and also informs the smartwatch as soon as the user has finished completing the self-report triggered by the server.

Just Accepted



(1) Onboarding screen (2) Chat screen (3) Affective slider screen (4) LimeSurvey screen

Fig. 3. Screenshots from the DyMand smartphone app. (1) Onboarding screen: These screens welcome the participants to the study. (2) Chat screen: The digital coach Pia or Pete chats with the participant on this screen. (3) Affective slider screen: The participants can choose their pleasure and arousal levels in the slider and submit them. (4) LimeSurvey screen: The self-reports, the end-of-the-dairy, baseline, and follow-up surveys are filled by the couples on this screen. See Appendix C for the translated text.

Just Accepted

4.4.1 Smartphone Self-Report Data Collection. The self-report (Figure 3) contains the Affective Slider [18] and a LimeSurvey questionnaire [55]. The Affective Slider is a digital emotion-measuring instrument that assesses emotions along the dimensions of valence and arousal [18]. The LimeSurvey questionnaire asked questions about social support, CDC, health behavior, and emotions. When a user needs to fill out a questionnaire prepared in LimeSurvey, the user sees a button that says "Fragebogen ausfüllen" (Start the survey). When the button is pressed, the link to the survey is opened. The link also includes relevant metadata, such as participant code that uniquely identifies the participant, which is needed to link survey answers to the specific partner in LimeSurvey. The self-report triggered after the 5 mins of smartwatch sensor data collect were set to expire after 4 minutes to ensure that partners started filling out the self-report within 4 minutes of getting the alert. Given that the self-report questions were about the previous 5 minutes of the couple's interaction when the sensor data was collected, the time limit helped to ensure the survey data matched the sensor data. The end-of-day diary questionnaire was set to expire after 45 minutes had more comprehensive questions on social support and CDC, healthy eating, medication adherence, and emotional well-being (e.g., short form of the PANAS self-report [81]). See [58] for details of the questions asked for the end-of-day questionnaire.

For the social support and CDC data that were collected right after 5 minutes of sensor data, each partner first had to respond to the question "Have you had any personal contact with your partner in the last five minutes?". If the response was "yes", there were subsequent questions, and if "no", the self-report ended. They then responded "yes" or "no" to the following statements about emotional support — for the partner with the disease (patient): "My partner has given me emotional support in the past five minutes" and for partners without the disease (caregiver): "I supported my partner emotionally in the past five minutes" and instrumental support — for patients:

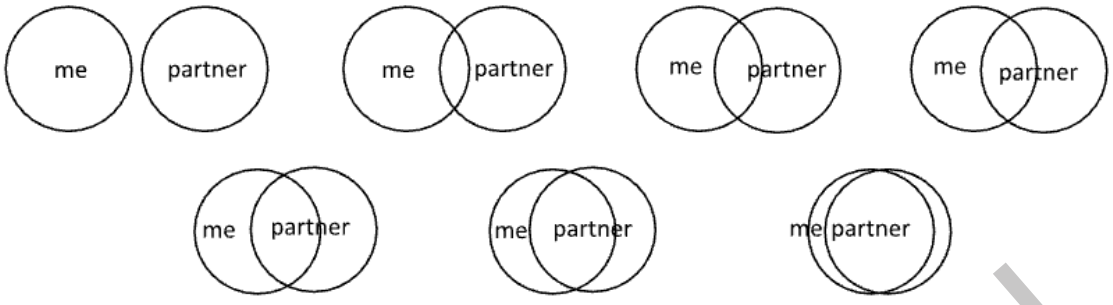


Fig. 4. Figures Options for How Partners Handled T2DM Together in the Last 5 Minutes

"My partner has given me practical support in the past five minutes" and for caregiver: "I supported my partner practically in the past five minutes." If the answer is yes, they are asked in what domain they received/provided support (physical activity, diet adherence, medication adherence, and other. These questions were adapted from Bolger et al. [29]. Finally, each partner had to select from a list of figures (Figure 4) the one that best expressed the handling of T2DM between them and their partner in the last five minutes (1 = complete separation of handling, 7 = handling completely together with your partner). This question captures the CDC construct in Aron et al. [14].

4.4.2 Smartphone Sensor Data Collection. When the Affective Slider is shown on the smartphone, a 3-second sensor data recording is made on the smartphone. The recording includes video from the front camera and continuous data from the ambient light sensor. For the video recording from the front camera, we used Android's MediaRecorder API [6]. For the ambient light sensor, we used the SensorManager API [10]. We registered this sensor with the parameter "SENSOR_DELAY_FASTEST" to get the sensor data as fast as possible. We stored the data locally on the phone which was retrieved when the couples returned their devices after they finished the study.

4.4.3 Continuously Running Smartphone Android App. For the DyMand system to function properly, the mobile app is required to run continuously (in the background) on the smartphone. As the smartwatch app was not expected to connect to the Internet on its own, if the connection to the smartphone app was lost, we could not relay the "recording done" message to the server. We implemented a Foreground service [4] in the mobile app which ran continuously and kept the app in a ready state throughout the study. In addition to the Foreground service, we changed the settings on the study phones that we gave to the couples such that the Android OS did not optimize the battery for our DyMand app. Even then, we found that in Nokia 6.1, one of the system apps (com.evenwell.powersaving.g3) still shut down our app after a few hours. We disabled this system app by using the Android Debug Bridge (ADB) and logging into the shell of the device.

4.4.4 Smartphone and Smartwatch Communication. For the communication between the DyMand apps in the smartphone and the paired smartwatch, we used the Wearable Data Layer API [9]. The following messages were sent between the smartphone and smartwatch applications.

(1) The weekday and weekend hours during which the couple is available for data collection [9] (chosen by the couple in the smartphone app) are sent from the smartphone app to the smartwatch app during the setup phase (section 4.6).

Just Accepted

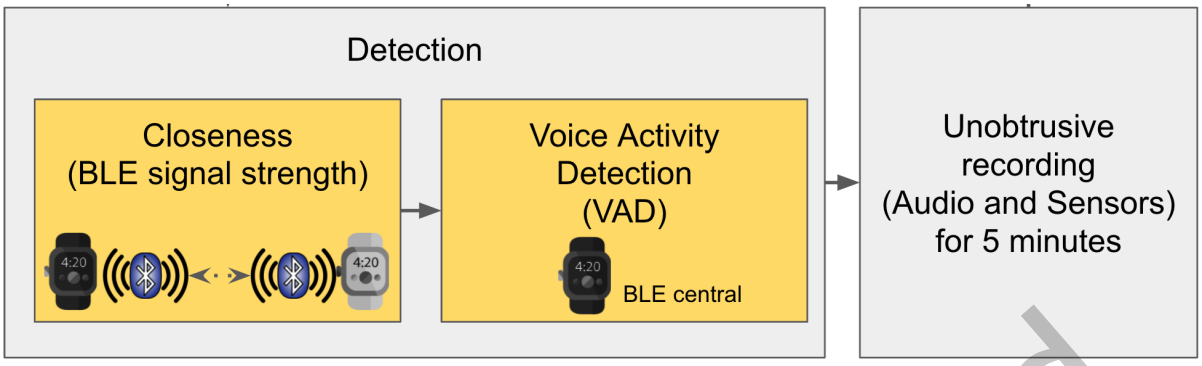


Fig. 5. Smartwatch App System Overview

(2) Text indicating that the smartwatch has finished collecting sensor data for 5 minutes is sent from the smartwatch app to the smartphone app throughout the study.
(3) Text indicating that the self-report on the smartphone has been completed is sent from the smartphone app to the smartwatch app throughout the study.
(4) Other messages for acknowledging received messages and logging are sent between the apps on the smartwatch and smartphone.

4.5 Smartwatch App

Similar to the smartphone app, we implemented a Foreground service [4] in the smartwatch app (Figure 5) that collected sensor data. The smartwatch app collected five minutes of the following sensor data once per hour within the morning and evening hours set by the couples: audio, heart rate, accelerometer, gyroscope, and ambient light. We collected a maximum of 5 minutes of data per hour for privacy reasons. Hence, to optimize the quality of data collected within that hour and to ensure that we recorded the most relevant 5 minutes of data (when partners are interacting), rather than triggering data collection at random or scheduled times, the app collected data when 1) the partners were physically close and 2) when there was speech. Our algorithm uses a two-step process. First, the app determines physical closeness using the BLE signal strength between the smartwatches (section 4.5.1) and checks if the signal strength is within a certain threshold, which corresponds to a distance estimate (section 4.5.2). Second, the app determines if the partners are speaking by using a voice activity detection (VAD) machine-learning algorithm, which is implemented on the smartwatch (section 4.5.3). In the case in which this condition of physical closeness and speaking is not met in the hour, the app triggers a backup recording in the last 15 minutes of the hour.

After the 5-minute recording ends, the watch vibrates and sends a trigger to the smartphone app via the Wearable Data Layer API [9] to bring up the self-report for that partner to complete. If the smartwatch does not receive a message from the smartphone app within 2 mins indicating that the self-report has been started, it gives another vibration alert. If once more, within the next 2 minutes, there is still no response about the start or completion of the self-report, it implies the self-report was not completed. Consequently, the app deletes the audio and attempts to trigger another sensor data collection and self-report for the rest of the hour. Doing this ensured that we collected data with matching sensor and self-report samples. The app also ensured that there were at least 20 minutes between subsequent data collection to reduce the burden of the partners completing the self-reports.

Just Accepted

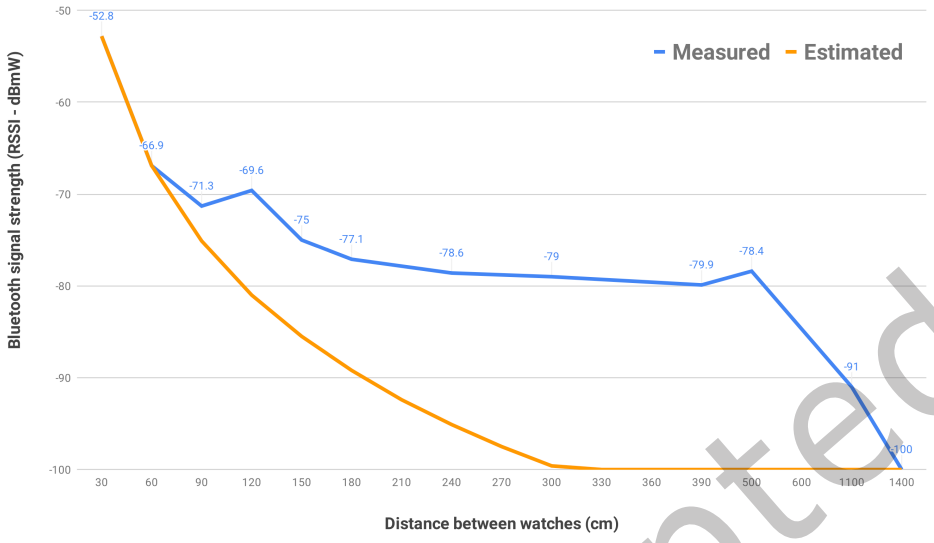


Fig. 6. Plot of RSSI between two smartwatches versus distance for real-world experiment vs. the theoretical expectation that the signal strength increases exponentially when the devices get closer and closer [48].

4.5.1 Physical Closeness Estimation. We used the BLE signal strength between the two smartwatches — one acting as the central (does BLE scanning) and the other acting as the peripheral (does BLE advertising) — to estimate the physical closeness of the partners. We conducted an experiment to measure the signal strength based on the distance between the two watches. In a lab setting, we placed two smartwatches at the same level and without any barrier between them. With one BLE peripheral watch fixed at a position, we placed the BLE central watch at a given distance from the former and we measured the signal strength. We varied the distances between the watches and repeated the experiment 10 times. We averaged the values and plotted them as shown in Figure 6 along with the theoretical expectation that the signal strength increases exponentially when the devices get closer and closer [48]. Our measurement showed that the signal strength is proportional to the closeness of the smartwatches but not perfect as in any BLE measurement. We chose a threshold of -80dB in the app which covers a range typically less than 5 meters which we assumed should adequately capture the distance between partners when they are interacting. We acknowledge that in the field, the presence of objects such as walls and furniture will affect the signal strength which we did not factor into the experiment. Nonetheless, our goal was not to have a very precise distance versus signal strength mapping but an approximate value to use for closeness between partners.

4.5.2 Physical Closeness Detection. For the DyMand smartwatch app, we implemented a BLE service in the smartwatch app for real-time physical closeness detection. At the start of each of the hours assigned for data collection, the peripheral smartwatch continuously advertises its universally unique identifier (UUID). The UUID is created based on the couple ID which is set uniquely for a couple during the onboarding of the study (e.g., P001 for the supporting partner and Z001 for the patient) (section 4.6). The central smartwatch scans with the UUID and when it finds the corresponding peripheral smartwatch, it checks the signal strength. If the signal strength is greater than -80 dB, then it tries to connect to the peripheral smartwatch. After a successful connection, both BLE

Just Accepted

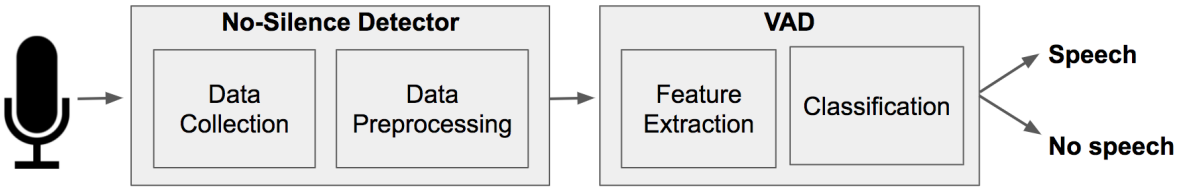


Fig. 7. System Architecture of VADLite

services acknowledge it by sharing a message through the Bluetooth channel. After this, the central smartwatch does voice activity detection (section 4.5.3). If the signal strength is less than -80 dB, the BLE scanning waits until the strength breaches the threshold. If there are any problems when the devices connect, the BLE scanning is reset and started from the beginning. Both the central and peripheral BLE services are attached to the Foreground service so that they can run in the background continuously.

4.5.3 Voice Activity Detection. We developed VADLite, a lightweight, open-source voice activity detection (VAD) system that runs in real-time on the smartwatch (Figure 7). An offline and online evaluation of VADLite using real-world data showed better performance than WebRTC's VAD, a popular open-source VAD system. VADLite is a 2-stage system consisting of a no-silence detector and a voice activity detector (see Appendix D and [25] for details of the development and evaluation of the system).

For the DyMand smartwatch app, the VAD component is triggered after the physical closeness detection. It records and samples audio at 8KHz and processes them in 5-second chunks by first performing no-silence detection and then speech detection. If it classifies the segment as speech, the VAD audio recording is stopped, and then sensor data collection is started on the central device (since it runs the VAD component) and a signal is sent to the peripheral device to immediately start sensor data collection.

4.5.4 Smartwatch Sensor Data Collection. The data collection component of the smartwatch app collected audio, heart rate, accelerometer, and gyroscope for 5 minutes. For the audio data, we used the MediaRecorder API [6] available in Android. We collected 16-PCM mono at 44.1 kHz. We set the output format of the audio file as ".wav" which is a lossless file format. All the other sensor data were collected using the SensorManager API [10]. We registered these sensors with the parameter "SENSOR_DELAY_FASTEST" to get the sensor data as fast as possible. The data collected was stored locally on the smartwatches and retrieved after the devices were returned.

4.5.5 DyMand App Checker. In early deployments, we realized that the smartwatch app could sometimes be shut down by the Wear OS system after an error, and our implementation for the app to restart itself was not always reliable. Consequently, we developed a smartwatch app — DyMand App Checker — to continuously check each hour if the DyMand smartwatch app is running and then start it if it is not running.

4.5.6 Logging in Smartwatch. To ensure that we understood how well the system was performing, we included various logs in the smartwatch app that were saved in files on the smartwatch, similar to [23]. In particular, we had 1) configuration logs at the time of setup, 2) hourly logs between the setup time of the devices and the start of the study, 3) hourly logs during the 7-day study period, 4) continuous log of the BLE signal strength between the two smartwatches, 5) hourly app function logs and 6) error logs as and when they happened. See Appendix E.2 for details of these.

Just Accepted

4.6 Setup Components of Smartwatch and Smartphone Apps

We implemented setup components of the smartphone and smartwatch apps for the partners to set up their devices, to ensure all aspects of the apps were working correctly before they take the devices away, and also for them to experience the process of sensor and self-report data collection. The smartphone app collects personal data and also the hours that the partners indicate for data collection. This information is transferred to the smartwatch app during the setup of the smartwatch after the "send" button is pressed on the smartphone. If the watch receives it, it shows a screen saying that the configuration process is successful. Otherwise, it needs to be sent from the smartphone again through the same button. The communication between a paired smartwatch and smartphone is explained in detail in Section 4.4.4. The smartwatch app collects information such as the couple ID (e.g., 001, 002) for logging purposes and creates a unique UUID (section 4.5.2) that can be used by the BLE services for advertising and scanning, the color of their smartwatch for the app to know if the watch belongs to the patient or supporting partner (by default, our smartwatch app chooses white to be the BLE peripheral and the black to be a BLE central device), receives the data collection hours from the smartphone, and collects a voice sample for 1 minute from the partner by asking them to read some text on a paper. Next, the BLE central smartwatch starts scanning and the BLE peripheral smartwatch starts advertising. After a successful connection, a message is shared via this Bluetooth connection between the smartwatches, and the recording of the sensor data starts in the watches, completes after 5 minutes, triggers self-report on the smartphone, and then triggers the end-of-day dairy.

5 Deployment: Field Study

After various internal pilot tests of the DyMand system, we deployed it in a field study with couples. We ran the DyMand study between 2019 and 2021 with heterosexual romantic couples from the German-speaking part of Switzerland in which one partner had T2DM (Figure 8) [58]. In total, we collected data from 13 couples aged 47 to 81 years, with a mean age of 68 (SD = 9) resulting in a total of 1,019 5-minute samples of sensor data (85 hours) and 598 corresponding completed self-report data.

The study was advertised in various places including hospitals, magazines, local newspapers, and the diabetes association in Switzerland. Interested couples completed a web-based questionnaire to screen them for the inclusion and exclusion criteria, and collect socio-demographic information. Those who met the eligibility criteria were able to pick a date for a baseline assessment at the laboratory of our university. During this session, both partners received comprehensive information about the study, signed the informed consent form, and completed a web-based questionnaire that captured constructs of interest at baseline that were not assessed daily.

They also received instructions on the study and then trained research assistants helped them to set up their devices and pair the corresponding smartphone and smartwatch (see Section 4.6 for details of the setup process). Each partner was given a smartwatch and smartphone running the DyMand apps and they were instructed to have all devices with them every day for 7 days from getting up until going to bed. To prevent mistakes from one partner accidentally using the other partner's watch and phone, one set of phones and watches had black covers and the other set had white covers. The patient was given the white set and the supporting partner was given the black set. The partners picked the hours during which we could collect data from them. They could choose any period from 4 am to 11 am for the morning hours and from 4 pm to 11 pm for the evening hours. During the weekend, only the early morning hours and late evening hours were set (e.g., from 6 am to 10 pm). With this procedure, privacy aspects were addressed by primarily focusing on situations, in which the couples spent time together and thus reducing the number of audio recordings during the day of weekdays when chances are higher that subjects are working or moving around in public places.

We collected data from their daily life for 7 consecutive days starting the next Monday after their visit until the following Sunday night. The DyMand system collected sensor and self-report data as described in previous

Just Accepted

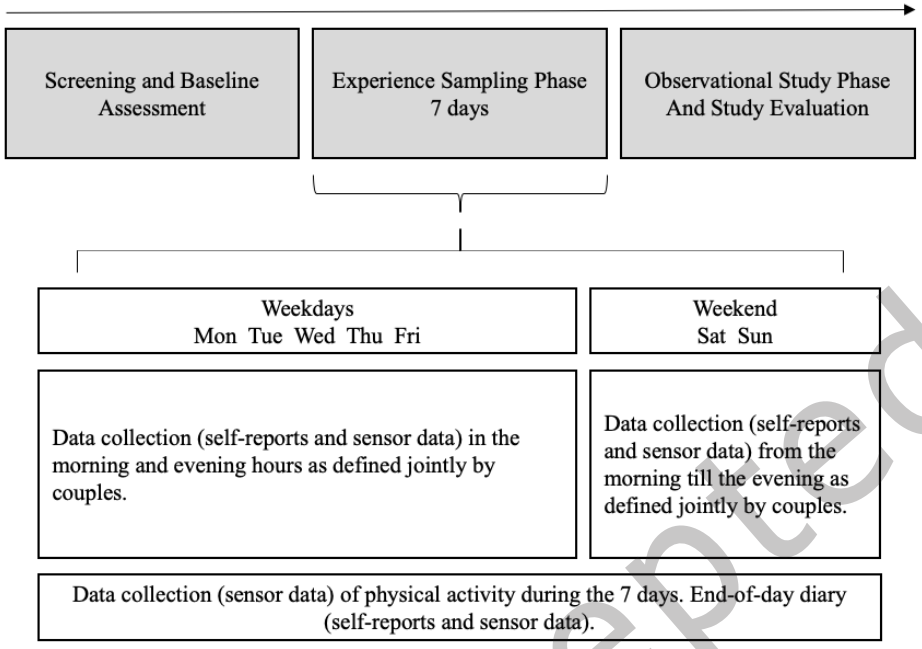


Fig. 8. Overview of DyMand study [58]

sections. During the study, we had a process for monitoring the study to ensure that the system was working correctly and to enable us to intervene if needed. We had a spreadsheet with cells corresponding to the hours during which we should have received data. Research assistants checked the DyMand MobileCoach server daily to see if the self-reports were triggered and completed, with each cell receiving one of the following values: 0 — no self-report trigger received, 1 — self-report trigger received, but no survey completed 2 — trigger received and survey completed. Additionally, we had a sheet that we used to track the details of issues and complaints that the couples communicated to the research team either via email or calls. We collected the following relevant information to understand and solve any problems that came up: subject id, device id, does the issue involve the watch, does the issue involve the phone, detailed description of the issues and pictures if available, date and time of issue, where they were when the issue happened, was the watch switched on, was the phone switched on, were the two watches physically close together, were the phone and watch physically close together, was there a connection to the digital coach indicated by the top-right icon on the smartphone app being green, were the phone and watch connected, does the phone have internet, and are both apps on the watch running.

There are significant ethical and privacy concerns in such a system and study since we collect audio which is sensitive data, and more so in the context of couples' interactions with the likelihood of speech about private topics. We took several measures as follows. First, our study received ethical clearance from the cantonal ethics committee of the Canton of Zurich, Switzerland (Req-2017_00430). Second, we ensured that we collected a maximum of 5 minutes of audio per hour in order not to record a significant percentage of the couples' everyday life. Consequently, even if the system triggers multiple recordings in the hour, the app always deletes all but the last one before the end of the hour. Third, to protect the privacy of subjects not taking part in the study, we asked subjects to wear a tag that we give them to indicate to others around that recording may be happening and

Just Accepted

that they may be recorded. Finally, after subjects returned their devices, we gave them the option to listen to and request the deletion of any audio samples without any explanation before the study team could listen to the audio files. Similar measures have been used in previous studies [61, 70] and have proven adequate to safeguard the privacy of study subjects and others not taking part in the study.

6 Evaluation

We evaluated the DyMand system regarding its technical performance and usability. For technical performance, the DyMand apps on the smartwatch and smartphone performed hourly logs (section 4.5.6) of relevant system performance metrics such as whether data collection started and completed, and whether self-report data was triggered, started, and completed. Furthermore, we annotated the audio data with relevant information such as whether it contained speech, and whether there was a conversation between partners for the algorithm-triggered recordings and the backup recordings.

6.1 Self-report and Sensor Data Collection

We investigated the percentage of the total expected number of sensor data and corresponding self-report data that was collected. Given that couples indicated the hours during which we could collect data, we estimated the total expected number for each couple and then summed them for all couples. Furthermore, given that the DyMand smartwatch app may not always be running due to circumstances such as the device being off because the couples did not charge them or the app crashing because of various errors such as in the BLE stack, a better metric would be the percentage of the expected number of sensor data and self-report data that was collected for hours when the app was running. For example, our logs showed that one couple did not turn on the smartwatch for the duration of the study. Hence, the app did not run and could not collect any data. Using the app's hourly logs, we estimated the number of hours that the app was running. We then used the following log status data: Was the sensor data collected? Was the self-report triggered? Was the self-report started? and Was the self-report completed? We computed sums of these log events for each partner, and then for all couples (Table 1). Additionally, we computed relevant percentages as shown in Table 2.

Our DyMand system collected 73.2% of the total expected sensor data and triggered 73.2% of the total expected self-reports. Considering only the case where the app was running, these percentages become 99.1% which shows that the system adequately triggered data collection as expected. Additionally, partners started 60.6% and completed 58.7% of the triggered self-reports. This percentage is not very high yet understandable for the context of this study. Partners had a maximum of 4 minutes to start the self-report after the trigger. The self-report was dismissed if it was started after this time. There are several reasons partners may not have started the self-report such as they did see or hear the alert because they were not wearing the watch, or they did not have the phone close by them. For those who saw the alert and attempted to start the self-report, some partners complained that there was a delay in the self-report loading on the smartphone due to internet connection issues (see Section E.4) and may have affected the start of the self-report in time.

6.2 Capturing Partners' Conversation Moments

We investigated how well the DyMand system captured speech and conversation between partners. We define a conversation as the presence of speech from both male and female partners in that audio. We had 3 trained research assistants annotate all valid audios (i.e., were not corrupted and hence playable, N=1014) by providing 'yes' or 'no' to relevant information such as: does it contain speech, did the male partner speak, did the female partner speak, and was there a conversation between partners. We automatically extracted the information about whether each audio was triggered or a backup recording by looking at the timestamp. Audios collected anytime between the 44th minute and the end of the hour (e.g., between 6:44 am and 7 am) are backup recordings. All audio

Just Accepted

Table 1. Expected and actual number of sensor and self-report data that were collected

Data	# of Samples
Total expected # of sensor and self-report	1392
Expected # of samples with the app running	1028
# of sensor data collected	1019
# of self-reports triggered	1019
# of self-reports started	618
# of self-reports completed	598

Table 2. Percentages of the expected number of sensor and self-report data that were collected

Data	Percentage (%)
% of total expected sensor data that was collected	73.2
% of expected sensor data with the app running that was collected	99.1
% of total expected self-report triggers that happened	73.2
% of total expected self-report triggers with the app running that happened	99.1
% of triggered self-report that were started	60.6
% of triggered self-report that were completed	58.7

Table 3. Number of audio samples with speech and conversation between partners broken down by triggered and backup recordings

Data Field	# of Samples
Total audios	1014
Audios with speech	791
# of triggered recordings	277
Triggered recordings with speech	256
# of backup recordings	737
Backup recordings with speech	535
Conversation between partners for all audios	538
Conversation between partners for triggered recording	215
Conversation between partners for backup recording	323

Just Accepted

samples collected before then were triggered by our algorithm. We show the sum of audio status information for all partners in Table 3 and calculate relevant percentages in Table 4.

For triggered audios, 92.4% contained speech, which shows VADLite has good performance in capturing speech in the real world considering prior work has shown that open-source VAD tools perform poorly on real-world speech data collected with smartwatches [25, 54]. Furthermore, 77.6% of triggered recordings had a conversation between partners in comparison to 43.8% of backup recordings which had a conversation between partners. This result shows that our smartwatch-based approach of physical closeness plus speech detection was better at capturing couples' conversation/interaction moments (77.6%) in comparison to using random or scheduled times per hour in our study (43.8%). It is difficult to have a direct comparison to other works. Nonetheless, in the following work [77], random triggering (10 times a day for one week) of self-report data collection among couples resulted in data for which partners were together (a proxy for interaction) in 39% of the triggers. Also,

Table 4. Percentage of audio samples with speech and conversation between partners broken down by triggered and backup recordings

Data	Percentage (%)
% of total audios with speech	78
% of triggered recordings with speech	92.4
% of backup recording with speech	72.6
% of total audios with conversation between partners	53.1
% of triggered recordings with conversation between partners	77.6
% of triggered recordings where one partner spoke	88.1
% of backup recordings with conversation between partners	43.8

another work that used the EAR triggered data collection among couples at a scheduled time sequence (every 9 minutes during waking hours) resulting in data for which patients and spouses talked (a proxy for partners' speech), on average, 47.9% and 45.0% of their waking hours [70]. These percentages are lower than the 77.6% from the DyMand system.

There are some likely reasons for the absence of conversation between partners in triggered recordings — the outstanding 22.4%. Firstly, our definition and evaluation metric of conversation between partners is strict — both male and female partners spoke in that 5-minute audio. In actuality, the partners may have been having a conversation but one partner may not have spoken in the specific 5-minute period in which we collected data, resulting in a "no" for conversation between partners. When we relax that definition to be either partner spoke, the percentage increases to 88.1%. Secondly, our physical closeness detection approach assumes that the partners are always wearing the smartwatch which was not always the case. For example, if the two smartwatches are left together on a table with the radio or TV in the background, a recording was triggered but it did not contain a conversation between partners. One way to address this issue is to include a pre-step to estimate if the device is being worn using the accelerometer or heart rate data as an example. Thirdly, another edge case that happened was partners sitting together and watching TV but not having a conversation. In this case, the recording was triggered because of the physical closeness and speech from the TV but the audio did not contain speech from the partners. One way to address this issue is to use an extra step for speaker identification to check if the speech is from either partner. Doing this will entail collecting a voice sample from both partners at setup, and then automatically creating and storing an acoustic fingerprint of each partner on the watch, and checking if there is a match after the voice activity detection.

6.3 Battery Consumption

We evaluated the smartwatch's battery consumption using hourly log data collected before the start of the study, when the application was only logging but not running the features of the application (idle state), and during the study when the application was running all features (active state). For both periods, we computed the mean hourly battery consumption, restricting the analysis to intervals during which the smartwatch was not being charged. In the idle state, the smartwatch had an average battery depletion rate of 1.6% per hour (std=0.25) based on data from a total of 1,250 hours of logging across 13 couples' smartwatches. This rate increased to 2.7% per hour (std=0.66) based on data from 2,287 hours of logging data in the active state, corresponding to a 70.5% increase in consumption relative to the idle state. Based on these rates, the estimated continuous operating duration of the smartwatch before requiring recharging is approximately 62.8 hours (2.6 days) in the idle state and 36.8 hours (1.5 days) in the active state. Review of the charging logs indicated that participants typically

Just Accepted

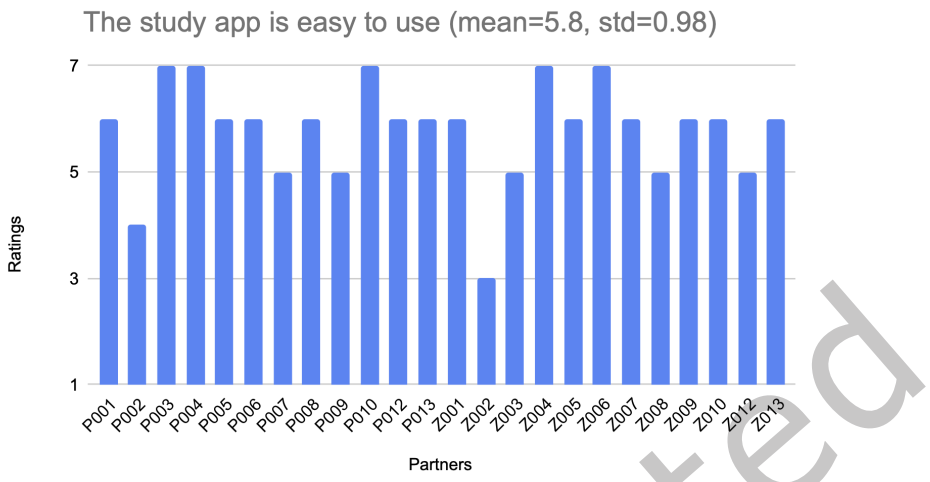


Fig. 9. Usability result showing partners' responses to the statement "The study app was easy to use" on a 7-point Likert scale ranging from strongly disagree (1) to strongly agree (7)

charged the smartwatch overnight while sleeping. Consequently, in practice, the device rarely reached a fully depleted battery state during the study.

6.4 Usability Results

At the end of the 7-day field study, partners completed a self-report on their experience with the DyMand apps. They responded to the statement "The study app was easy to use" on a 7-point Likert scale ranging from strongly disagree (1) to strongly agree (7). As shown in Figure 9, most ratings were high with a mean of 5.8 (std=0.98, N=24), which shows the DyMand system was easy to use. Also, partners wrote open-ended responses about potential areas of improvement for the DyMand system. One key recurring theme was that there was a delay in the self-report showing up as mentioned before. Another common suggestion was to reduce the number of questions.

7 Chronic Disease Management Insights

DyMand's algorithm-based trigger of sensor and self-report data collection of partners' conversation moments based on partners' interaction enables analysis and insights into various constructs in the context of chronic disease management. In this work, we describe how the audio and self-report data enabled an understanding of partners' social support behavior, CDC, and diabetes discussions in situ on a fine-grained scale (5 minutes of partners' interactions). Prior works have derived well-being insights and use cases of other DyMand sensor data, such as heart rate for the automatic detection of each partner's emotional well-being [26] and heart rate linkage among partners when giving or receiving social support [65].

7.1 Social Support and Diabetes Discussion from Audios

Four trained research assistants (RAs) coded the 5-minute audio recordings (992 audios coded; rest used for training RAs) for the presence of social support and discussion of diabetes management [65] using approaches similar to Wang and Repetti [80] (for support) and Robbins et al., [70] (for chronic disease discussion). For social

Just Accepted

support, they had to indicate "yes" or "no" for the presence or absence of support. Furthermore, if support was present, they had to indicate if it was instrumental or emotional. Supportive interactions were defined as any opportunity to provide instrumental support or emotional support. To be coded as support, the help needed to be provided for meaningful or complex activities that required some burden on the part of the provider. Relatedly, each support occurrence was coded for the type of support being provided. Instrumental support entailed help with practical problems and tasks, such as assistance with chores or the provision of information to help handle a task-oriented problem (e.g., figuring out the fastest driving route to the doctor's office). Emotional support included provisions of comfort, encouragement, advice, or guidance of an emotional nature, for example, listening empathetically to a spouse's frustrations about work. RAs had an interrater reliability score of 0.63 (intraclass correlation coefficient) based on 15% of recordings.

For discussion of diabetes management, the RAs had to indicate "yes" or "no" if diabetes is mentioned. In particular, they had to code it as "yes" if they inferred from the conversation that the participant is talking about diabetes or diabetes-related issues, e.g., going to the doctor, treatment, blood sugar levels, taking medication, symptoms of hypo- or hyperglycemia (tiredness, feeling shaky, confusion, headache, weakness), health behaviors to manage diabetes (diet, exercise). When talking about health behaviors or symptoms, the link to the disease needed to be specifically mentioned.

Within conversations between partners, 5.2% of couples' conversation moments (n=591) were coded as containing social support with 67.7% being instrumental and 32.3% being emotional. This result from this small sample suggests that support may not be a major part of couples' interactions in daily life even in the context of chronic disease management and is consistent with the work of Wang and Repetti [80] who found 4.3% of couples' interactions captured via at-home video recording (morning and evening) of 30 healthy couples over 4 days contained support where the majority was instrumental (82%) and 18% for emotional. Furthermore, partners discussed diabetes management of the patient in 2.4% of their conversation moments, indicating that it is a small percentage of partners' conversations. This result is consistent with that of Robbins et al. [70] who found that partners discuss disease management in 5% of conversations (random audio samples from 112 couples, managing breast cancer of one partner in daily life). This feasibility study demonstrates how certain constructs in the context of couples' diabetes management can be assessed in daily life and in situ, facilitated by the DyMand system, which captures partners' conversation moments in a privacy-preserving manner — once each hour, as opposed to all-day intrusive audio or video recording used by other works.

7.2 Social Support and CDC from Self-Reports

We analyzed the self-report responses related to social support and CDC, described in Section 4.4.1. Patients indicated that they received support (practical or emotional) in 67.7% of couples' interactions (n=257). Caregivers indicated that they provided support (practical or emotional) in 71.8% of couples' interactions (n=216). Patients had an average of 5.2 out of 7 in their assessment of the handling of T2DM between them and their partners (n=257, std=2.2). Caregivers had an average of 5.1 in their assessment of the handling of T2DM between them and their partner (n=216, std=2.0).

These findings suggest that partners frequently self-reported supportive interactions during the observed time frames. The high rates of support underscore the active engagement of both partners in the caregiving dynamic and point to a generally supportive relational context. However, these perceptions contrast sharply with the observational data derived from coded audio recordings of the same interactions, where only 5.2% of conversations included observable instances of social support. Possible reasons for this discrepancy are described in [65]. Social desirability bias could lead to participants over-reporting provided and received support. Also, interactions that partners perceive as support may not be apparent to coders who are not familiar with the couples'

Just Accepted

unique interaction dynamics. An exit interview that specifically explores the differences in the observational and self-reported data about social support will be useful in the future to understand the discrepancy.

Similarly, both groups reported moderately high levels of collaborative disease management, with patients averaging 5.2 (SD = 2.2) and caregivers averaging 5.1 (SD = 2.0) on a 7-point scale assessing shared handling of T2DM. These ratings indicate a shared perception that the illness is being managed together rather than individually.

However, slight discrepancies between patient and caregiver reports may point to potential differences in perception. Caregivers reported higher rates of support provision than patients reported receiving, which may suggest that some supportive efforts may go unnoticed or be interpreted differently by each partner. Additionally, patients' slightly higher ratings of collaborative disease management may reflect a greater sense of shared involvement from their perspective, or alternatively, a caregiver's perception of greater individual responsibility.

Taken together, these results highlight both the strength of supportive dynamics and the presence of potentially subtle misalignments in how they are experienced. It is important to note that these insights are preliminary, given that we had a small sample of couples and we analyzed audio samples from only 5 mins within each hour for the hours in which we collected data.

The DyMand system enables the collection and analysis of data that provides valuable insights that could inform potential interventions to improve partners' social support and CDC in the context of chronic disease management. This insight has important implications for the design of mHealth tools aimed at supporting couples managing chronic illness. Specifically, these tools could include features that encourage shared reflection on support exchanges, promote visibility of caregiving efforts, and facilitate real-time communication about emotional and practical needs. Interactive prompts, dyadic check-ins, or shared goal-setting functionalities could help align perceptions and reinforce collaborative management. By addressing both behavioral and perceptual aspects of support, mHealth interventions can more effectively strengthen dyadic coordination, improve relational understanding, and ultimately support better disease outcomes.

8 Limitations and Future Work

When issues happened during the study, we sometimes did not have enough information to infer the cause until the devices were returned. Even though we had various key logs that would have helped, we did not implement a way to transfer the logs from the watch to the phone, and then to our server for monitoring. The DyMand system was already quite complex and we had time constraints; hence, we decided not to invest time and effort into implementing this feature. One key implementation challenge was the fact that the watch did not have a direct internet connection and hence, any such data transfer would have had to happen through the phone and then to the server, further introducing potential points of failure. To enhance better error analysis in the field, in the future, the DyMand system should be extended to transfer the logs from the watch to the phone and then to an external server. Also, smartwatches with internet connection could be used in the future, so the data can be sent directly to the server.

Our algorithm for capturing couples' interaction moments, though performed well, did not address some edge cases as described in the evaluation section (section 6.2). Future work will update the algorithm to check if the partners are wearing the smartwatch as part of the closeness and speech detection check by leveraging other sensor data such as acceleration. Furthermore, we will implement a speaker identification method to check if the speech is from either of the partners and avoid being triggered by other people or speech from the radio or TV. This implementation will entail training a speaker identification model to extract a speech embedding from each partner during the setup and use it for comparison in real-time on the smartwatch.

Our health analyses were based on a small sample of couples (n=13). The COVID-19 pandemic affected data collection, making it difficult to get a larger sample. Furthermore, only couples 12 and 13 had their data collection

Just Accepted

during the pandemic, which may have impacted the reported and observed behavior of conversation moments, social support, and CDC. Hence, the findings from this study need to be replicated on a larger sample of couples to evaluate their generalizability.

Our DyMand system currently only collects relevant sensor and self-report data but it does not perform real-time recognition of constructs that are relevant for understanding couples' chronic disease management such as emotional well-being, social support, and CDC, frequency of discussion of diabetes management. Future work will develop approaches for the automatic detection of these constructs and then augment the DyMand system with such capabilities. These will reduce the manual effort involved in coding these constructs and also enable near real-time insights for timely interventions.

Our approach of capturing only 5 minutes of audio data per hour, though privacy-perserving contrained how much depth of insight that could be obtained from partners' interactions in daily life. Future work could explore (1) collecting longer duration (e.g., 20 minutes) or several 5-minute samples per hour, and (2) starting data collection after interaction detection and only stop the recording after no interaction is detected anymore (3) implementing machine learning approaches to continuously extract low-level acoustic features from the data, but delete the audio, or high-level insights as described in the previous paragraph, to then have a complete insight into partners' interactions while preserving their privacy.

The DyMand system is generic in that it is suitable not only for couples' diabetes management but also for studies in the context of related diseases such as hypertension, or mental health disorders, in which sensor and self-reports about emotional well-being and health behavior are relevant for disease management and health intervention designs. Given that it is open source, it can be extended or adapted, and our method for capturing partners' conversations/interactions can be used by other researchers to develop similar apps to collect data to understand various constructs among other dyadic constellations such as friendships, and sibling and parent-child dyads. Also, another potential research use case is to better understand communication patterns in situ and performance measures of teams in organizations.

9 Conclusion

In this work, we developed, deployed, and evaluated the DyMand smartwatch and smartphone system that captures couples' dyadic interactions in daily life in the context of chronic disease management. It consists of a smartwatch app, and a smartphone app, built on top of the MobileCoach platform that collects sensor and self-report data that are relevant for chronic disease management on couples' interaction/conversation moments. We deployed DyMand in a 7-day field study and collected 85 hours of data from 13 heterosexual romantic couples from the German-speaking part of Switzerland. Key challenges affected the system's performance and usability such as software errors, poor Internet connectivity, and long self-report questionnaires. Nonetheless, our evaluations showed that the system had good performance in triggering the collection of the expected number of sensor and self-report data and capturing couples' conversation moments, and it was easy to use. Our audio analysis showed that 5.2% of couples' conversation moments contained social support with 67.7% being instrumental and 32.3% being emotional, and partners discussed diabetes management of the patient 2.4% of the time. The self-report data showed that patients received support in 67.7% of couples' interactions and caregivers provided support in 71.8% of couples' interactions. Furthermore, patients had an average of 5.2 (out of 7) in their assessment of the handling of T2DM between them and their partners and caregivers had an average of 5.1. DyMand could be used by social, clinical, or health psychology researchers to understand the social dynamics of couples that are managing chronic diseases in everyday life to inform the development and delivery of behavioral interventions. Our system could be customized and extended to be used in other contexts besides chronic disease management such as couples' daily dynamics more broadly, workplace interactions, and other dyad constellations such as parent-child and sibling-sibling or roommate dyads.

Just Accepted

References

[1] 2022. Apache Tomcat. Retrieved March 28, 2022 from https://tomcat.apache.org/
[2] 2022. ASPSMS. Retrieved March 28, 2022 from https://www.aspsms.com/en/
[3] 2022. Deepstream. Retrieved March 28, 2022 from https://deepstream.io/
[4] 2022. Foreground services. Retrieved March 28, 2022 from https://developer.android.com/guide/components/foreground-services
[5] 2022. Letsencrypt. Retrieved March 28, 2022 from https://letsencrypt.org/
[6] 2022. MediaRecorder. Retrieved March 28, 2022 from https://developer.android.com/reference/android/media/MediaRecorder
[7] 2022. MongoDB. Retrieved March 28, 2022 from https://www.mongodb.com/
[8] 2022. NGINX. Retrieved March 28, 2022 from https://www.nginx.com/
[9] 2022. Send and sync data on Wear OS. Retrieved March 28, 2022 from https://developer.android.com/training/wearables/data/data-layer
[10] 2022. SensorManager. Retrieved March 28, 2022 from https://developer.android.com/reference/android/hardware/SensorManager
[11] 2022. Using polar products in research. Retrieved March 28, 2022 from https://www.polar.com/en/science/research-tools
[12] 2022. Vaadin Framework. Retrieved March 28, 2022 from https://vaadin.com/
[13] Tuka AlHanai and Mohammad Ghassemi. 2017. Predicting latent narrative mood using audio and physiologic data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.
[14] Arthur Aron, Elaine N Aron, and Danny Smollan. 1992. Inclusion of other in the self scale and the structure of interpersonal closeness. *Journal of personality and social psychology* 63, 4 (1992), 596.
[15] Nilanjan Banerjee, Sharad Agarwal, Paramvir Bahl, Ranveer Chandra, Alec Wolman, and Mark Corner. 2010. Virtual compass: relative positioning to sense mobile social interactions. In *Pervasive Computing: 8th International Conference, Pervasive 2010, Helsinki, Finland, May 17-20, 2010. Proceedings 8*. Springer, 1–21.
[16] Rummana Bari, Md Mahbubur Rahman, Nazir Saleheen, Megan Battles Parsons, Eugene H Buder, and Santosh Kumar. 2020. Automated detection of stressful conversations using wearable physiological and inertial sensors. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies* 4, 4 (2020), 1–23.
[17] Isabella C Bertschi, Fabienne Meier, and Guy Bodenmann. 2021. Disability as an interpersonal experience: a systematic review on dyadic challenges and dyadic coping when one partner has a chronic physical or sensory impairment. *Frontiers in psychology* 12 (2021).
[18] Alberto Betella and Paul FMJ Verschure. 2016. The affective slider: A digital self-assessment scale for the measurement of human emotions. *PloS one* 11, 2 (2016), e0148037.
[19] George Boateng. 2020. Towards a wearable system for assessing couples' dyadic interactions in daily life. In *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*. 208–211.
[20] George Boateng, John A Batsis, Ryan Halter, and David Kotz. 2017. ActivityAware: an app for real-time daily activity level monitoring on the amulet wrist-worn device. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 431–435.
[21] George Boateng, John A Batsis, Patrick Proctor, Ryan Halter, and David Kotz. 2018. GeriActive: Wearable app for monitoring and encouraging physical activity among older adults. In *2018 IEEE 15th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. IEEE, 46–49.
[22] George Boateng and David Kotz. 2016. StressAware: An app for real-time stress monitoring on the amulet wearable platform. In *2016 IEEE MIT Undergraduate Research Technology Conference (URTC)*. IEEE, 1–4.
[23] George Boateng, Vivian Genaro Motti, Varun Mishra, John A Batsis, Josiah Hester, and David Kotz. 2019. Experience: Design, development and evaluation of a wearable device for mHealth applications. In *The 25th Annual International Conference on Mobile Computing and Networking*. 1–14.
[24] George Boateng, Prabhakaran Santhanam, Janina Lüscher, Urte Scholz, and Tobias Kowatsch. 2019. Poster: DyMand–An Open-Source Mobile and Wearable System for Assessing Couples' Dyadic Management of Chronic Diseases. In *The 25th Annual International Conference on Mobile Computing and Networking*. 1–3.
[25] George Boateng, Prabhakaran Santhanam, Janina Lüscher, Urte Scholz, and Tobias Kowatsch. 2019. VADLite: an open-source lightweight system for real-time voice activity detection on smartwatches. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*. 902–906.
[26] George Boateng, Xiangyu Zhao, Malgorzata Speichert, Elgar Fleisch, Janina Lüscher, Theresa Pauly, Urte Scholz, Guy Bodenmann, and Tobias Kowatsch. 2026. "Are You Okay, Honey?": Recognizing Emotions Among Couples Managing Diabetes in Daily Life Using Multimodal Real-World Smartwatch Data. *Sensors* 26, 10 (2026). https://doi.org/10.3390/s26103141
[27] Guy Bodenmann. 2005. Dyadic coping and its significance for marital functioning. (2005).
[28] Niall Bolger and David Amarel. 2007. Effects of social support visibility on adjustment to stress: Experimental evidence. *Journal of personality and social psychology* 92, 3 (2007), 458.

Just Accepted

[29] Niall Bolger, Adam Zuckerman, and Ronald C Kessler. 2000. Invisible support and adjustment to stress. *Journal of personality and social psychology* 79, 6 (2000), 953.
[30] Ciro Cattuto, Wouter Van den Broeck, Alain Barrat, Vittoria Colizza, Jean-François Pinton, and Alessandro Vespignani. 2010. Dynamics of person-to-person interactions from distributed RFID sensor networks. *PloS one* 5, 7 (2010), e11596.
[31] Tanzeem Choudhury and Alex Pentland. 2003. Sensing and modeling human networks using the sociometer. In *Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings.* IEEE, 216–222.
[32] Sheldon Cohen, Lynn G Underwood, and Benjamin H Gottlieb. 2000. Social support measurement and intervention: A guide for health and social scientists. (2000).
[33] Nathan Eagle and Alex Pentland. 2006. Reality mining: sensing complex social systems. *Personal and ubiquitous computing* 10 (2006), 255–268.
[34] Nathan Eagle, Alex Pentland, and David Lazer. 2009. Inferring friendship network structure by using mobile phone data. *Proceedings of the national academy of sciences* 106, 36 (2009), 15274–15278.
[35] Tiantian Feng, Amrutha Nadarajan, Colin Vaz, Brandon Booth, and Shrikanth Narayanan. 2018. TILES audio recorder: an unobtrusive wearable solution to track audio activity. In *Proceedings of the 4th ACM Workshop on Wearable Systems and Applications*. ACM, 33–38.
[36] Denzil Ferreira, Vassilis Kostakos, and Anind K Dey. 2015. AWARE: mobile context instrumentation framework. *Frontiers in ICT* 2 (2015), 6.
[37] Andreas Filler, Tobias Kowatsch, Severin Haug, Fabian Wahle, Thorsten Staake, and Elgar Fleisch. 2015. MobileCoach: A novel open source platform for the design of evidence-based, scalable and low-cost behavioral health interventions: overview and preliminary evaluation in the public health context. In *2015 Wireless Telecommunications Symposium (WTS)*. IEEE, 1–6.
[38] Claudio Forlivesi, Utku Günay Acer, Marc van den Broeck, and Fahim Kawsar. 2018. Mindful interruptions: a lightweight system for managing interruptibility on wearables. In *Proceedings of the 4th ACM Workshop on Wearable Systems and Applications*. ACM, 27–32.
[39] Ekin Gedik and Hayley Hung. 2018. Detecting conversing groups using social dynamics from wearable acceleration: Group size awareness. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 4 (2018), 1–24.
[40] Michele Girolami, Fabio Mavilia, and Franca Delmastro. 2020. Sensing social interactions through BLE beacons and commercial mobile devices. *Pervasive and Mobile Computing* 67 (2020), 101198.
[41] Jasara N Hogan, Alexander O Crenshaw, Katherine JW Baucom, and Brian RW Baucom. 2021. Time spent together in intimate relationships: Implications for relationship functioning. *Contemporary Family Therapy* 43, 3 (2021), 226–233.
[42] John F Horton, Pro Stergiou, Tak S Fung, and Larry Katz. 2017. Comparison of Polar M600 optical heart rate and ECG heart rate during exercise. *Med Sci Sports Exerc* 49, 12 (2017), 2600–2607.
[43] Syed Monowar Hossain, Timothy Hnat, Nazir Saleheen, Nusrat Jahan Nasrin, Joseph Noor, Bo-Jhang Ho, Tyson Condie, Mani Srivastava, and Santosh Kumar. 2017. mCerebrum: a mobile sensing software platform for development and validation of digital biomarkers and interventions. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*. 1–14.
[44] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. 2003. A practical guide to support vector classification. (2003).
[45] Hayley Hung, Gwenn Englebienne, and Laura Cabrera Quiros. 2014. Detecting conversing groups with a single worn accelerometer. In *Proceedings of the 16th international conference on multimodal interaction*. 84–91.
[46] International Diabetes Federation Europe (IDF). 2022. IDF member(s) in Switzerland. Retrieved 7th May, 2022 from https://idf.org/our-network/regions-members/europe/members/161-switzerland.html
[47] Masumi Iida, Mary Ann Parris Stephens, Karen S Rook, Melissa M Franks, and James K Salem. 2010. When the going gets tough, does support get going? Determinants of spousal support provision to type 2 diabetic patients. *Personality and Social Psychology Bulletin* 36, 6 (2010), 780–791.
[48] Zhu Jianyong, Luo Haiyong, Chen Zili, and Li Zhaohui. 2014. RSSI based Bluetooth low energy indoor positioning. In *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 526–533.
[49] Alexander Karan, Robert C Wright, and Megan L Robbins. 2017. Everyday emotion word and personal pronoun use reflects dyadic adjustment among couples coping with breast cancer. *Personal Relationships* 24, 1 (2017), 36–48.
[50] Cynthia M Khan, Mary Ann Parris Stephens, Melissa M Franks, Karen S Rook, and James K Salem. 2013. Influences of spousal support and control on diabetes management through physical activity. *Health Psychology* 32, 7 (2013), 739.
[51] Noah Klugman, Veronica Jacome, Meghan Clark, Matthew Podolsky, Pat Pannuto, Neal Jackson, Aley Soud Nassor, Catherine Wolfram, Duncan Callaway, Jay Taneja, et al. 2018. Experience: Android Resists Liberation from Its Primary Use Case. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. 545–556.
[52] Tobias Kowatsch, Theresa Schachner, Samira Harperink, Filipe Barata, Ullrich Dittler, Grace Xiao, Catherine Stanger, Florian v Wangenheim, Elgar Fleisch, Helmut Oswald, et al. 2021. Conversational agents as mediating social actors in chronic disease management involving health care professionals, patients, and family members: multisite single-arm feasibility study. *Journal of medical Internet research* 23, 2 (2021), e25060.
[53] Tobias Kowatsch, Dirk Volland, Iris Shih, Dominik Rüegger, Florian Künzler, Filipe Barata, Andreas Filler, Dirk Büchter, Björn Brogle, Katrin Heldt, et al. 2017. Design and evaluation of a mobile chat app for the open source behavioral health intervention platform

Just Accepted

MobileCoach. In *International Conference on Design Science Research in Information System and Technology*. Springer, 485–489.
[54] Daniyal Liaqat, Robert Wu, Andrea Gershon, Hisham Alshaer, Frank Rudzicz, and Eyal de Lara. 2018. Challenges with real-world smartwatch based audio monitoring. In *Proceedings of the 4th ACM Workshop on Wearable Systems and Applications*. 54–59.
[55] LimeSurvey Project Team / Carsten Schmitz. 2012. *LimeSurvey: An Open Source survey tool.* LimeSurvey Project, Hamburg, Germany. http://www.limesurvey.org
[56] Fannie Liu, Mario Esparza, Maria Pavlovskaia, Geoff Kaufman, Laura Dabbish, and Andrés Monroy-Hernández. 2019. Animo: Sharing biosignals on a smartwatch for lightweight social connection. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 1 (2019), 1–19.
[57] Timothy J Loving and Richard B Slatcher. 2013. Romantic relationships and health. *The Oxford handbook of close relationships* (2013), 617–637.
[58] Janina Lüscher, Tobias Kowatsch, George Boateng, Prabhakaran Santhanam, Guy Bodenmann, and Urte Scholz. 2019. Social Support and Common Dyadic Coping in Couples' Dyadic Management of Type II Diabetes: Protocol for an Ambulatory Assessment Application. *JMIR research protocols* 8, 10 (2019), e13685.
[59] Janina Lüscher and Urte Scholz. 2018. Soziale unterstützung. *Psychologie in der Gesundheitsförderung. Bern, Germany: Hogrefe* (2018).
[60] Aleksandar Matic, Venet Osmani, Alban Maxhuni, and Oscar Mayora. 2012. Multi-modal mobile sensing of social interactions. In *2012 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops*. IEEE, 105–114.
[61] Matthias R Mehl, Megan L Robbins, and Fenne große Deters. 2012. Naturalistic observation of health-relevant social processes: The Electronically Activated Recorder (EAR) methodology in psychosomatics. *Psychosomatic Medicine* 74, 4 (2012), 410.
[62] Dirk Merkel. 2014. Docker: lightweight linux containers for consistent development and deployment. *Linux journal* 2014, 239 (2014), 2.
[63] Daisy Miller and J Lynne Brown. 2005. Marital interactions in the process of dietary change for type 2 diabetes. *Journal of Nutrition Education and Behavior* 37, 5 (2005), 226–234.
[64] Daniel Olguín Olguín, Benjamin N Waber, Taemie Kim, Akshay Mohan, Koji Ara, and Alex Pentland. 2008. Sensible organizations: Technology and methodology for automatically measuring organizational behavior. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39, 1 (2008), 43–55.
[65] Theresa Pauly, Janina Lüscher, Lea Olivia Wilhelm, Melanie Alexandra Amrein, George Boateng, Tobias Kowatsch, Elgar Fleisch, Guy Bodenmann, and Urte Scholz. 2024. Using Wearables to Study Biopsychosocial Dynamics in Couples Who Cope With a Chronic Health Condition: Ambulatory Assessment Study. *JMIR mHealth and uHealth* 12 (2024), e49576.
[66] Mashfiqui Rabbi, Shahid Ali, Tanzeem Choudhury, and Ethan Berke. 2011. Passive and in-situ assessment of mental and physical well-being using mobile sensors. In *Proceedings of the 13th international conference on Ubiquitous computing*. 385–394.
[67] Maija Reblin, Richard E Heyman, Lee Ellington, Brian RW Baucom, Panayiotis G Georgiou, and Susan T Vadaparampil. 2018. Everyday couples' communication research: Overcoming methodological barriers with technology. *Patient education and counseling* 101, 3 (2018), 551–556.
[68] Tuula-Maria Rintala, Pia Jaatinen, Eija Paavilainen, and Päivi Åstedt-Kurki. 2013. Interrelation between adult persons with diabetes and their family: a systematic review of the literature. *Journal of family nursing* 19, 1 (2013), 3–28.
[69] Megan L Robbins, Elizabeth S Focella, Shelley Kasle, Ana María López, Karen L Weihs, and Matthias R Mehl. 2011. Naturalistically observed swearing, emotional support, and depressive symptoms in women coping with illness. *Health Psychology* 30, 6 (2011), 789.
[70] Megan L Robbins, Ana María López, Karen L Weihs, and Matthias R Mehl. 2014. Cancer conversations in context: naturalistic observation of couples coping with breast cancer. *Journal of Family Psychology* 28, 3 (2014), 380.
[71] Theodore F Robles, Richard B Slatcher, Joseph M Trombello, and Meghan M McGinn. 2014. Marital quality and health: A meta-analytic review. *Psychological bulletin* 140, 1 (2014), 140.
[72] Michael J Rohrbaugh, Matthias R Mehl, Varda Shoham, Elizabeth S Reilly, and Gordon A Ewy. 2008. Prognostic significance of spouse we talk in couples coping with heart failure. *Journal of consulting and clinical psychology* 76, 5 (2008), 781.
[73] Nina Rottmann, Dorte Gilså Hansen, Pia Veldt Larsen, Anne Nicolaisen, Henrik Flyger, Christoffer Johansen, and Mariët Hagedoorn. 2015. Dyadic coping within couples dealing with breast cancer: A longitudinal, population-based study. *Health Psychology* 34, 5 (2015), 486.
[74] Ralf Schwarzer and Nina Knoll. 2010. Social support. *Health psychology, 2nd ed.* (2010), 283–293.
[75] Amber J Seidel, Melissa M Franks, Mary Ann Parris Stephens, and Karen S Rook. 2012. Spouse control and type 2 diabetes management: moderating effects of dyadic expectations for spouse involvement. *Family relations* 61, 4 (2012), 698–709.
[76] Vedran Sekara and Sune Lehmann. 2014. The strength of friendship ties in proximity sensor data. *PloS one* 9, 7 (2014), e100915.
[77] Laura Sels, Jed Cabrieto, Emily Butler, Harry Reis, Eva Ceulemans, and Peter Kuppens. 2019. The occurrence and correlates of emotional interdependence in romantic relationships. *Journal of personality and social psychology* (2019).
[78] Adela C Timmons, Brian R Baucom, Sohyun C Han, Laura Perrone, Theodora Chaspari, Shrikanth S Narayanan, and Gayla Margolin. 2017. New frontiers in ambulatory assessment: Big data methods for capturing couples' emotions, vocalizations, and physiology in daily life. *Social Psychological and Personality Science* 8, 5 (2017), 552–563.

Just Accepted

[79] Adela C Timmons, Theodora Chaspari, Sohyun C Han, Laura Perrone, Shrikanth S Narayanan, and Gayla Margolin. 2017. Using multimodal wearable technology to detect conflict among couples. *Computer* 50, 3 (2017), 50–59.
[80] Shu-wen Wang and Rena L Repetti. 2016. Who gives to whom? Testing the support gap hypothesis with naturalistic observations of couple interactions. *Journal of Family Psychology* 30, 4 (2016), 492.
[81] David Watson, Lee Anna Clark, and Auke Tellegen. 1988. Development and validation of brief measures of positive and negative affect: the PANAS scales. *Journal of personality and social psychology* 54, 6 (1988), 1063.
[82] Katharina Weitkamp, Fabienne Feger, Selina A Landolt, Michelle Roth, and Guy Bodenmann. 2021. Dyadic Coping in Couples Facing Chronic Physical Illness: A Systematic Review. *Frontiers in psychology* 12 (2021).
[83] Wikipedia contributors. 2022. Transport Layer Security — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Transport_Layer_Security&oldid=1079821520 [Online; accessed 28-March-2022].

Just Accepted

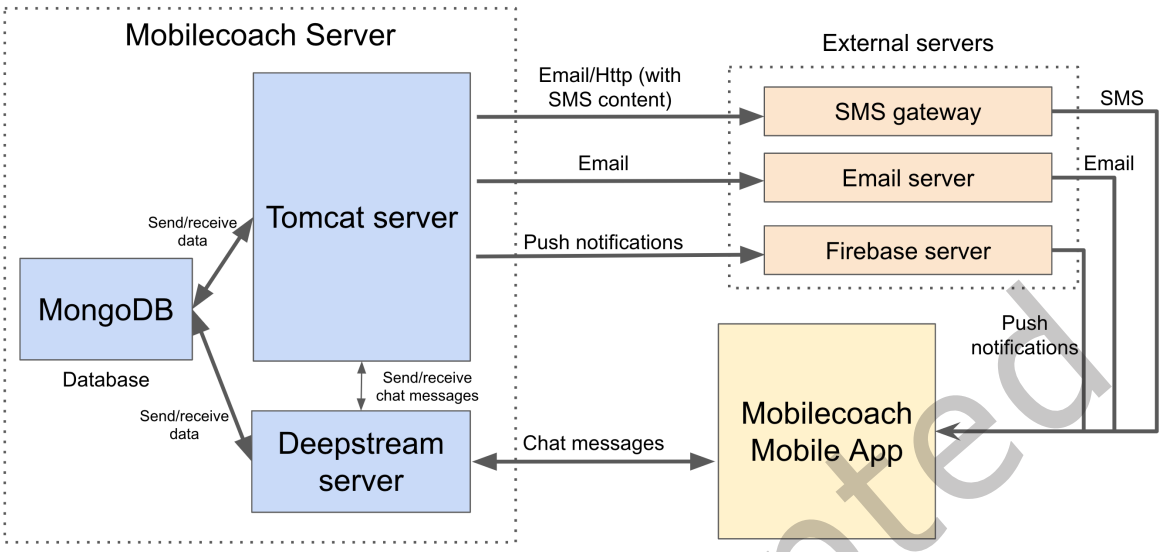


Fig. 10. DyMand MobileCoach Server and its functions

Just Accepted

A MobileCoach Backend

The MobileCoach server (Figure 10) of the DyMand system is hosted in an Ubuntu 16.04 virtual machine in our university's network. The framework provides a Docker-based setup. The Docker platform [62] performs OS-level virtualization and helps in the portability of software. The server setup of MobileCoach comes with containers running Tomcat [1], DeepStream [3], MongoDB [7] and Nginx [8]. Tomcat provides a Java web server environment and it serves the MobileCoach Designer (section B) which is built using the Vaadin framework [12]. The intervention engine of MobileCoach processes the rules and schedules the messages. It connects with the user interface (UI) of the MobileCoach Designer and communicates (i.e., sends/receives the messages) with the mobile app using the real-time server DeepStream. DeepStream allows clients and backend services to synchronize data. The database is managed by MongoDB, which is a NoSQL database program. The communication between the server and the mobile app happens through secure SSL [83] encrypted connections. We used the certificate authority Let's Encrypt [5] for setting up the free SSL certificates. We performed regular security upgrades and monitored the server. In addition to sending messages via chat, the MobileCoach framework was configured to send SMS or emails. For emails, we used the SMTP server provided by our university. For SMS, we used ASPSMS [2] service. Additionally, for the questionnaires, we hosted a LimeSurvey setup in the DyMand server. LimeSurvey [55] is a free and open-source online survey tool we used to create our questionnaires. We picked LimeSurvey because it provides free community editions and detailed setup instructions, it is easily customizable, and we could host it on our DyMand server so that the data can reside within our universities network.

B MobileCoach Intervention Designer

The MobileCoach intervention designer has user interface (UI) elements (Figure 11) where dialogs can be created with messages using multiple answer option types. The conversational agent (PIA or PETE) serves these dialogues to the partners. There are three main modes of rule execution in MobileCoach designer as follows: (1) daily

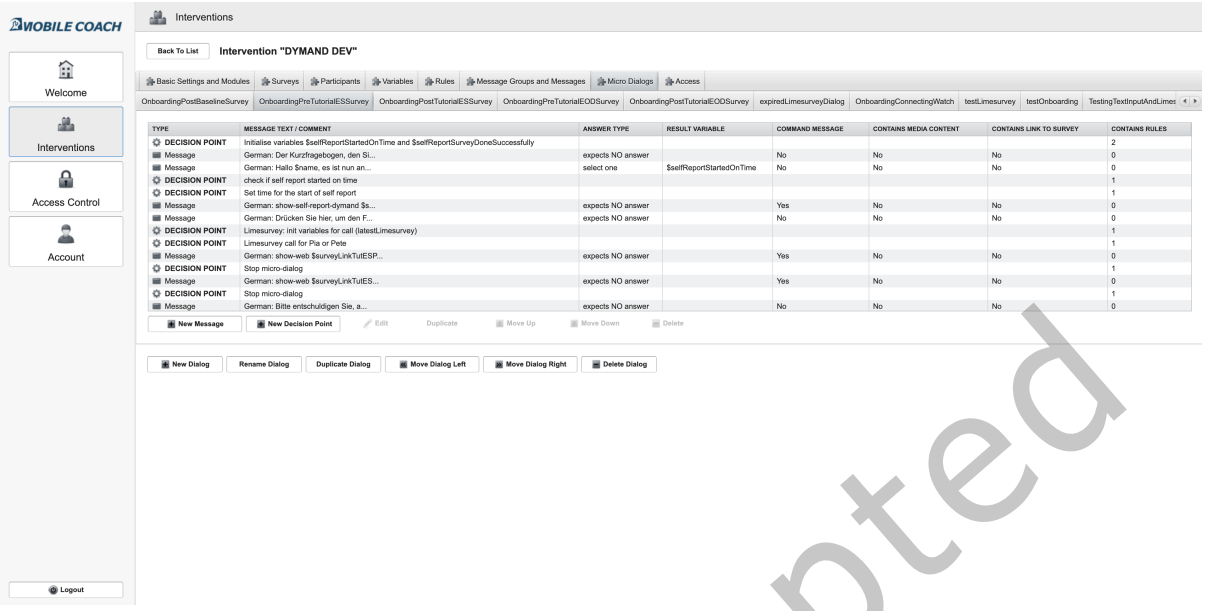


Fig. 11. MobileCoach Intervention Designer

execution (once every day at midnight), (2) almost-continuously execution (every 5 minutes), and (3) event-triggered execution (e.g., events can be triggered passively by sensor inputs or manually by user interactions in the mobile application). Each dialogue is connected to either of these rules. When conditions of rules are satisfied, then corresponding dialogues are sent. The conditions are designed based on system and user-defined variables available in the MobileCoach Designer. Each participant has a copy of these variables, which defines the state of the participant. The MobileCoach designer also provides an interface through which study participants and their states (based on the variables) can be monitored. This interface also allows the export of all messages sent and received with timestamps, and the most recent variable values for each participant for data analyses. For the DyMand system, we designed the following dialogues:

(1) Onboarding dialogue is sent when the app is opened for the first time during the setup of the study apps. It is used to get the personal information from each partner (e.g., to approach them by their nicknames), to complete the baseline questionnaire before the start of the study, to deliver relevant study information, and help with onboarding the participants for the study.
(2) Reminder messages in the form of text messages (SMS) were sent to the personal phones of the couples on Sunday evening and again on Monday morning reminding them to take their study smartphones and wear the study smartwatches directly after getting up to prepare them for the 7-day data collection.
(3) Self-report dialogue requests participants to fill out a self-report questionnaire after the sensor data collection on the smartwatch. The self-report was a LimeSurvey questionnaire that asked questions about social support, CDC, health behavior, and emotions (e.g., short form of the PANAS self-report [81]). This dialogue used the event-triggered rule execution mode which was initiated when the sensor data collection was successful. In particular, the MobileCoach designer provides various commands to be sent to the mobile app. These commands can have different purposes in the app. For example, the command "show-web <https://abc.de> <Button-name>", will show a button with the name "Button-name" which

Just Accepted

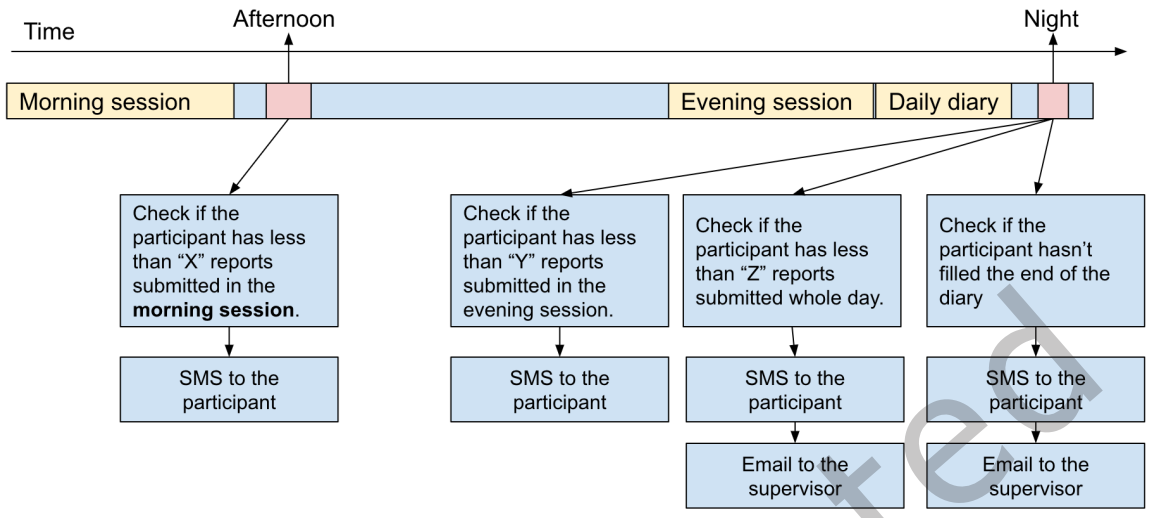


Fig. 12. DyMand Escalation mechanism: The values X, Y, and Z are calculated every day based on the number of hours the participants specify that they are available for the study in the morning or the evening period and the percentages of self-reports that we set as thresholds for escalation.

Just Accepted

when pressed would open the link “https://abc.de” in a new screen within the app. These links in the DyMand system (Figure 2) were our pre-configured questionnaire from LimeSurvey which additionally provided choices to enter JavaScript code. We added codes in each of our surveys that enabled the app to close the survey screen and continue the dialog with the conversational agent. Using JavaScript code, we added timers to the surveys. They were set to expire after 4 minutes to ensure that partners started filling out the self-report within 4 minutes of getting the alert. Given that the self-report questions were about the previous 5 minutes of the couple’s interaction when the sensor data was collected, the time limit helped to ensure the survey data matches the sensor data.

(4) End-of-day diary questionnaire dialogue requests the partners to fill a self-report survey at the end of the day with more comprehensive questions on social support and CDC, healthy eating, medication adherence, and emotional well-being [58]. Similar to the implementation of the self-report dialogue, they were set to expire after 45 minutes.
(5) Follow-up dialogue was sent when the 7-day field study was finished. This dialogue asked partners to fill out a follow-up survey about their study experience [58].
(6) Escalation messages were designed as both text messages and emails that were sent to the partners and study supervisors when the partners were less adherent to completing self-reports in the study. As shown in Figure 12, there were two checks done by the system automatically every day during the 7-day study period — one at 2 pm, and another one at the end of the day, after the end-of-day dairy is sent. As shown in Figure 12, when the partners do not fill at least 60% percent of the self-reports in the morning or the evening period, a reminder SMS is sent to them. Additionally, if the end-of-day dairy is not completed or if the total number of the self-reports done for the whole day was less than 30%, then the participant received an SMS and the study supervisor was sent an email additionally to follow up with the couple via phone call.

C Translation of App's Text

See Table 5 for English translation of the German text in the screenshot of the app in Figure (3)

Table 5. German and English translations used in the study

Category	**German Text**	**English Translation**
Onboarding	DyMand Studie	DyMand Study
Onboarding	Herzlich Willkommen zu unserer Studie "Partnerschaft und Diabetes". Bitte drücken Sie Weiter, um fortzufahren.	Welcome to our study "Partnerschaft und Diabetes". Please press forward to continue.
Onboarding	Weiter	Continue
Chat Screen	Ja	Yes
Chat Screen	Bitte bestätigen Sie nochmals mit ja, dass die Synchronisierung vollzogen ist. Wenn nein gedrückt wird, werden die Zeitangaben erneut versendet.	Can you please confirm again with yes, that the configuration is sent? If no is pressed, the configuration will be resent.
Chat Screen	Vielen Dank. Das Smartphone und die Smartwatch sind nun für die erste Studienphase bereit!	Thank you very much. The smartphone and smartwatch are now ready for the first phase of the study!
Chat Screen	Es ist nun an der Zeit, den Kurzfragebogen für diese Stunde auszufüllen. Möchten Sie fortfahren?	It is now time for you to fill the self report for this hour. Do you want to continue?
Chat Screen	Ja, ich möchte fortfahren	Yes, I'd like to continue
Affective Slider	Bitte teilen Sie uns mit den beiden Schiebern mit, wie Sie sich in den letzten 5 Minuten gefühlt haben. Denken Sie nicht zu lange nach, folgen Sie einfach Ihrem Bauchgefühl.	Please tell us with the two sliders how you felt during the last 5 minutes. Do not think too long, just follow your gut feeling.
Affective Slider	Wie müde bzw. wach fühlten Sie sich in den letzten 5 Minuten?	How tired or awake have you felt in the past five minutes?
Affective Slider	Wie unglücklich bzw. glücklich fühlten Sie sich in den letzten 5 Minuten?	How unhappy or happy have you felt in the past five minutes?
LimeSurvey	Hatten Sie in den letzten fünf Minuten mit Ihrem Partner/Ihrer Partnerin persönlichen Kontakt?	Have you had personal contact with your partner in the past five minutes?
LimeSurvey	Nein	No
LimeSurvey	Ja	Yes
LimeSurvey	Zurück	Back

Just Accepted

D VADLite Development and Evaluation

The no-silence detector computes the root mean square (RMS) of segments of the audio signal and marks them as non-silence if they are above a certain threshold. The voice activity detector consists of a feature extractor, and a machine-learning algorithm, which we trained to classify speech versus non-speech [25]. In particular, it



Fig. 13. Real-time testing of VADLite showing detection states of silence (left), speech (middle), and noise (right)

extracts mel-frequency cepstral coefficients and classifies speech versus non-speech audio samples using a linear Support Vector Machine (SVM). An SVM is a classifier that constructs a high-dimensional hyperplane to separate data of different classes [44]. SVM selects a hyperplane that maximizes the distance to the nearest data points on either side of the hyperplane in the case of binary classification. We used a linear SVM because it is memory and computationally efficient when doing predictions. Prior work has used an implementation of linear SVM for real-time prediction on smartwatches for stress detection [22] and activity detection [20, 21].

To train VADLite, we ran a study that was approved by the ethics committee of our university. We collected lab and field audio data (16-PCM mono, 8KHz, 3.5 hours total) using the Polar M600 smartwatch from several people (at least 10 distinct individuals) at varying distances from the smartwatch. We annotated the audio samples as speech or non-speech. We preprocessed the data to remove silence segments by computing the RMS of each one-second time window and checking if the RMS value is below a certain threshold determined empirically. We then extracted frequency-based features — 13 MFCC features which have been widely used for VAD [38] and used 12 of them (excluding the 1st coefficient, which is the DC component) — over a time window of 25 ms non-overlapping time window. We normalized the features and used them to train a linear SVM to classify speech or non-speech. We then implemented the system to run in real-time on the smartwatch (Figure 13). The real-time implementation continuously collects audio and processes them in 1-second segments for no-silence detection and then voice activity detection.

We performed offline and online evaluations using the following metrics for evaluation: accuracy, speech hit rate (SHR), and false alarm rate (FAR). The SHR is the ratio of correctly detected speech frames to the total number of speech frames. By contrast, FAR is one minus the noise hit rate, where the noise hit rate is the ratio of correctly detected noise frames to the total number of noise frames. For offline evaluation, we split the data into train and test using about 70%-30% stratified split and performed 10-fold stratified cross-validation with hyperparameter tuning on the train data and evaluated on the test set. The model achieved 82.6% accuracy, 80.2% SHR, and 14.9% FAR. For online evaluation, we played 15-minute audio collected from a naturalistic context through a loudspeaker as the VADLite app performed real-time classification of the audio just like was done by Feng et al [35]. VADLite had SHR and FAR of 91.6% and 5.5% respectively. These results were better than WebRTC's VAD: offline accuracy, SHR, and FAR of 71.4%, 79.5%, and 37.5% respectively, and online SHR and FAR

Just Accepted

of 73% and 18% respectively. Furthermore, VADLite had an average processing time of 2 ms for each 25-ms frame and 76 ms for the total one-second duration. As a result, throughput was met since the frame processing time was less than the 25 ms segment duration. Likewise, the processing time for the whole duration was less than one second.

E Implementation Detailts

We describe various implementation details to allow for reproducibility.

E.1 Exception Handling in Smartwatch

We added try-catch statements in various parts of the code where they could be exceptions (e.g., writing text or data to a file, BLE scanning, etc.). Additionally, similar to [51], we included the "DefaultUncaughtExceptionHandler" to catch all uncaught exceptions. Our implementation of the Exception class spawns a thread that counts the number of exceptions in that hour and saves it (for later logging), saves the exception message (for logging), schedules a restart of the app with an AlarmManager for the next second, stops the current Foreground service, and shuts down the app.

E.2 Logging Details in Smartwatch

(1) The configuration log contained the dates and hours of data collection set by the couples which we used for post-study analysis.
(2) The before-study logs contained the log's timestamp, battery level, the number of days, hours, minutes, and seconds until the start of the study, and the number of exceptions that happened in the previous hour. We could use this data to infer various things such as whether the battery died before the study started and the partners forgot to charge it, or whether an error shut down the app.
(3) The during-study hourly logs contained several important fields. These included the following from the previous hour: timestamp of the log, battery level, date and number of times the BLE started advertising (peripheral device) or scanning (for the central device), date and number of times the device met the closeness condition, date and number of times of no-silence detection, date, and number of times of voice activity detection, date and number of times the two watches connected, date and number of times sensor data was collected, date and number of times self-report was triggered (1st and 2nd alert), started, and completed, whether the recording was a backup recording if the audio was discarded (because self-report was not completed), number and dates of errors, number of times and dates the app restarted, whether the internet was available on the smartphone, and amount of space remaining on the smartwatch. These data allowed us to assess the performance of the system (see Section 6).
(4) The hourly app's function logs contained various "print" statements we had in various functions in our code as well as the Wear OS system logs which our app logged and saved in a file. We did this to debug and better understand which code block may have caused any errors.
(5) The BLE log saved the signal strength between the devices throughout the continuous scanning during the hours of data collection. This data can be used to infer how much time partners spent together.
(6) The error logs were the exceptions (described in the previous section) that we logged. We used those to debug the app.

E.3 Errors

On our server, when there were too many user intent messages (messages that indicate a recording is done in the watch), DeepStream could not maintain the connection between the server and the client — a socket hang-up error. Though we could not find a technical solution to the problem, we managed to fix this by restarting the

Just Accepted

Docker container of DeepStream. As this happened only a couple of times during the study and as we found out the problem both times quickly during our periodic monitoring of the system, this error had a negligible effect on the data obtained. There were BLE connection issues, too. In the smartwatch, we found that after a few successful BLE connections, the smartwatches could not connect anymore with each other. We fixed this by periodically destroying and recreating the whole Bluetooth stack and not just the BLE services (scanning and advertising).

E.4 Challenges

Unavailability of Internet access sometimes resulted in self-reports not being shown on the smartphone in time as described before. This issue is a major limitation of the MobileCoach platform as the mobile app has to send a request to the server to then send a request back to the phone to then show the self-report on the screen introducing several points of failure. When this happens, the corresponding audio recording gets deleted as it is not valid for us to analyze without a corresponding self-report. The app then tries to record another sample and trigger another self-report, increasing the burden of self-report completion. A solution during the study would have been an offline trigger that syncs when the internet connection is restored. We did not anticipate this as we gave our participants phones with SIM cards for Internet access. Due to resource constraints, we could not change this later during the study. A future extension of the MobileCoach platform with an in-app offline trigger to show the self-report will circumvent these multiple pathways and will prevent this issue.

MobileCoach is designed such that the DeepStream server (Appendix A) syncs the messages designed in the intervention designer to the app only when the app is open and used by the user. After many "recording done" user-intent messages to the server, the server triggers many self-reports and when the app is opened after an inactive period, it takes some time for the messages to sync with the DeepStream server. This issue caused some delay for the most recent message to appear when the app was opened. As the solution would have required major customizations in the MobileCoach framework, we did not address this issue.

Making the smartwatch app run continuously (section 4.4.3) was quite challenging. In addition to the Foreground service and changing the battery optimization settings, one of the system apps in the Nokia 6.1 kept shutting down our app after a few hours. We spent a lot of time trying to figure out the root cause and fix it by disabling the respective system app (com.evenwell.powersaving.g3). This issue is important to note for future studies that plan to implement long-running background services on the Android platform.

Just Accepted