Master Thesis

Context–Sensitive Alerting and Reminding on Mobile Devices

Masterarbeit zur Erlangung des akademischen Grades

Master of Science

Verfasser: Christian Mayr

Vorgelegt am FH-Masterstudiengang MultiMediaTechnology, Fachhochschule Salzburg

Begutachtet durch: DI Dr. Simon Ginzinger MSc (Betreuer)

Salzburg, 24.11.2014

Eidesstattliche Erklärung

Hiermit versichere ich, Christian Mayr, geboren am **16.06.1984** in **Linz**, dass ich die Grundsätze wissenschaftlichen Arbeitens nach bestem Wissen und Gewissen eingehalten habe und die vorliegende Masterarbeit von mir selbstständig verfasst wurde. Zur Erstellung wurden von mir keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Ich versichere, dass ich die Masterarbeit weder im In- noch Ausland bisher in irgendeiner Form als Prüfungsarbeit vorgelegt habe und dass diese Arbeit mit der den BegutachterInnen vorgelegten Arbeit übereinstimmt.

Salzburg, am **24.11.2014**

Unterschrift

Christian Mayr	1210695015

Kurzfassung

Aktuelle technologische Geräte liefern große Unterstützung für Multitaskingumgebungen, die zu einer erhöhten Anzahl an Tasks führt, die zeitgleich ausgeführt werden können. Speziell Smartphones haben sich in den letzten Jahren zu Multifunktionsgeräten entwickelt, die eine Vielzahl an Aufgaben übernehmen können. Dadurch ist der Benutzer mit einer großen Anzahl von Benachrichtigungen konfrontiert, die ihm oder ihr mitteilen, dass etwas (mehr oder weniger) Wichtiges passiert ist. Je höher allerdings die Anzahl von solchen Benachrichtigungen ausfällt, desto höher ist auch die Wahrscheinlichkeit, dass man in einem unerwünschten Zeitraum unterbrochen wird.

In dieser Arbeit sammeln wir Ideen, um die Anzahl unerwünschter Unterbrechungen im Anwendungsfall eines Alarm– oder Erinnerungsservices auf mobilen Endgeräten zu verringern. Aktuelle Smartphones stellen eine Vielzahl an Sensoren und Zugriff auf Onlineservices (wie etwa Onlinekalender) bereit. Wir versuchen aufgrund dieser Daten kontextuelle Hinweise zu finden, um auf die empfundene Störung zukünftiger Unterbrechungen schließen zu können.

Im Rahmen einer Studie haben wir ein Programm für Androidgeräte entwickelt, welches unsere Teilnehmer in regelmäßigen Abständen unterbricht und Kontextdaten sammelt und speichert. Aus den Resultaten dieser Studie versuchen wir Muster zwischen kontextuellen Vorbedingungen und dem Störgrad des Alarms, der von unserer Implementierung erzeugt wird, zu finden. Auf Basis dieses Wissens liefern wir schließlich Vorschläge, um einen kontextsensitiven Erinnerungs– oder Alarmservice auf mobilen Geräten implementieren zu können.

Schlagwörter: Unterbrechung, Aufmerksamkeit, Context Aware Computing, Kontextdaten, Mobile Geräte, Entscheidungsmodelle

Abstract

Current technology advances highly support multitasking, leading to an increased amount of tasks which can be executed simultaneously. In the last years especially smartphones have emerged to multifunctional toolsets supporting a high variety of different tasks. As a result, the user is confronted with a high amount of notifications created by these devices, indicating, that something (more or less) important has happened. Therefore, with a higher amount of notifications the likelihood for being interrupted in unwanted moments increases.

In this thesis, we concentrate on gathering ideas for decreasing the amount of unwanted interruptions within the use case of alerting and reminding services on mobile devices. As nowadays smartphone devices contain a wide variety of sensors and deliver access to online services such as calendars, we try to extract contextual clues out of these data to predict the disruptiveness of future interruptions.

We conducted a study, where we developed a program for Android devices, which interrupts our participants at regular time intervals and collects and stores contextual data. Within the results of this study we try to find patterns between contextual preconditions and the disruptiveness of the alerts created by our application. Finally, we deliver suggestions to implement a reminding or alerting service on mobile devices based on this knowledge.

Keywords: *Interruption, Attention, Context Aware Computing, Contextual Data, Mobile Devices, Decision Models*

Contents

List of Abbreviations

am ante meridiem
API Application Programming Interface
APK Android Application Package
COI Cost of Interruption
CPM–GOMS Critical Path Method–Goals, –Operations, –Methods and –Selection Rules
CSA Context–Sensitive Alerts
e.g. exempli gratia
ECA Expected Cost of Alerting
ECI Expected Cost of Interruption
EVR Expected Value of Reminding
GOMS Goals, Operations, Methods and Selection Rules
GPS Global Positioning System
HCI Human–Computer Interaction
ID Identifier
IMEI International Mobile Equipment Identity
IMSM Interruption Management Stage Model
pm post meridiem
QR Quick Response
RCF Resource Competition Framework
SDK Software Development Kit
SIM Subscriber Identity Module
SQL Structured Query Language
UI User Interface
WPF Windows Presentation Foundation

1 Introduction

Technological advances lead to an increase of tasks, which can be performed by people simultaneously, yet their cognitive capabilities remain at the same level (McFarlane and Latorella 2002, 3). Designing software fitting those limitations has been tackled as a central human–computer interaction (HCI) design problem for the future (McFarlane and Latorella 2002, 4). As a specific example, smartphones have emerged to devices, which can not only be used for communication, but also as multifunctional toolsets by supporting multiple applications for a variety of purposes like gaming, browsing or listening to music (Leiva et al. 2012). For indicating that something (more or less) important has happened, these applications often take use of notifications to attract the user's attention (Poppinga, Heuten, and Boll 2014). Such strategies can be applied to implement a variety of software aiding the user in his or her daily life. Within this thesis, we concentrate on applications for alerting and reminding, specifically on mobile devices as they are widespread used (Pielot, Church, and Oliveira 2014).

When bringing up examples for benefits, within the medical area studies showed that "electronic reminders lead to short–term improvements of patients' adherence to chronic medication" (Vervloet et al. 2012). Also, it has been stated, that such an alerting and reminding system must be aware of the patient's medication–taking behaviors (Pavel et al. 2010). Pavel et al. mention, that most existing reminding and alerting systems are generating alerts at fixed time points and also outline the problems regarding this approach:

> "Although these have been shown to be useful, they frequently fail to achieve adherence for a variety of reasons, mostly associated with the patients' context and concurrent activities. For example, if the patient is involved in various concurrent activities such as sleeping, telephoning or is not near the medication dispenser when the alert is generated, he may not respond to the alert. In addition, if the alerting system is not aware whether or not the patient took the medication, it cannot generate follow up alerts." (Pavel et al. 2010)

We conclude, that it appears important to take the user's context in account and therefore find the right moment of interruption. All in all, the potential benefit of a reminder should be rated higher than the cost of interruption associated with transmitting the reminder (Kamar and Eric Horvitz 2011). Adamczyk and Bailey showed in their studies, that their "predicted best point for interruption consistently produced less annoyance, frustration, and time pressure, required less mental effort, and were deemed by the user more respectful of their primary task" (Adamczyk and Bailey 2004).

When taking the insights of the last paragraph into account, research in context–aware alerting and reminding can help to improve the overall user experience. Nowadays, mobile devices offer access to a wide variety of sensor data and online services. We attempt to use these features to gather clues about the user's context and to find opportune moments to interrupt. Therefore, we conduct a study collecting contextual data on Android devices.

Our participants are interrupted at regular time intervals and have to evaluate the disruptiveness of an alert created by our application. Within contextual data stored on an online database we try to find patterns between opportune and not opportune moments of interruption and the data collected. Also, recent research, which has been established in this area, will be compared to our approach.

Our final goal is to find contextual clues in connection to the disruptiveness of interruptions and to deliver strategies on how to use this knowledge to implement an alerting or reminding service on a mobile device. Within this introductive chapter we'll analyze current popular approaches taking the user's context in mind and point out the weaknesses we're aiming to overcome. After formulating the motivation of our research in more detail, we'll give an overview over the structure of this work and the main motivation behind each of the four main chapters of this thesis.

1.1 Context Aware Applications on Mobile Devices

Here, we'll outline a few applications, which have been released for mobile devices. First of all, we're aiming on analyzing context–aware assistants which have been integrated into the most popular mobile operating systems Android, iOS and Windows Phone. Next, we'll give a short overview over popular context–aware applications for the Android platform. We choose Android as target platform as our study is also focused on this system (reasons are outlined in Chapter 4.2).

1.1.1 Integrated Services

Google, Apple and Microsoft delivered solutions integrated in their operating systems, namely Google Now, Siri and Cortana. All of them use users' personal data to provide just–in–time context–aware services by delivering information a user might need, but didn't explicitly ask for (Pejovic and Mirco Musolesi 2014).

Google Now is for instance delivering information about the current weather, giving hints when to leave to a certain place depending on traffic situations or suggesting information to the user which is rated relevant towards his or her preferences[1]. Within this service, reminders or alerts can be triggered based on location or time.

Siri is following a comparable approach by delivering information tailored to the user's preferences or executing certain tasks like identifying songs or dictate messages[2]. In comparison to Google Now, Apple's approach is more focused on the idea of providing a personal assistant the user can communicate with by voice commands. As in Google Now, reminders can be set up and fired based on location and time.

Microsoft is also following the personal assistant approach with Cortana on Windows Phone – here the user can also execute tasks or gather information based on his or her

1. `https://www.google.com/intl/en/landing/now/#whatisit` (accessed on November 19, 2014)
2. `https://www.apple.com/ios/siri/` (accessed on November 19, 2014)

preferences[3]. When setting up reminders, they can also be bound to location and time. Additionally, they allow creating reminders bound to persons, firing when sending or receiving messages or phone calls from a specific contact[4].

All of these services are still in active development and their approaches justify the growing importance of context–awareness within mobile applications. But when focusing on our use case of alerting and reminding, we discover one flaw: All interruptions triggered by Google Now, Siri or Cortana are solely based on one contextual pre–condition (like time or location) but not judging the actual disruptiveness of the reminder. As a result, there's a risk that the user will be reminded at inopportune moments leading to a lowered efficiency of the reminder and a higher annoyance perceived in the user's everyday life.

1.1.2 Android Apps

Dayer et al. compared 160 different reminder Apps for medication adherence and evaluated them regarding their features. Of interest for us is their implication, that "identifying the reasons for nonadherence and developing a scale that assesses unintentional nonadherence would be a useful starting point toward effectively deploying app–based reminders" (Dayer et al. 2013). That's one point all these evaluated reminder Apps occur to lack and we're aiming to overcome by pursuing a more general approach via gathering clues about those mentioned reasons within patterns found in contextual evidence.

Within context–aware applications, we also evaluated Llama, which is a location aware mobile application for Android devices[5]. Here, the user can define areas and link actions to them based on events (e.g. mute the phone when being at home between 10 pm and 8 am). It is also possible to set up reminders based on such predefined conditions. Automagic follows a comparable approach, with the difference that such processes are defined in flow charts[6]. Tasker also allows the definition of sequences, but here tasks can be executed on a wider variety of contexts (which are application, time, date, location, event and gesture)[7]. All these tools are definitely useful for applying context–aware actions also for reminding or alerting, but they have one flaw in common we want to avoid: The user has to predefine each contextual possibility on his or her own.

We are following an approach, where the system judges about appropriate moments of interruptions on its own and tailors these decisions based on the user's individual schedule. Summing up, we couldn't find any evidence for a released mobile application fulfilling the demands specified within our study.

3. `http://www.microsoft.com/en-us/mobile/campaign-cortana/` (accessed on November 19, 2014)
4. `http://www.windowsphone.com/en-us/how-to/wp8/cortana/remind-me-cortana` (accessed on November 19, 2014)
5. `http://kebabapps.blogspot.co.at/p/llama-handling-guide-instructions.html` (accessed on November 19, 2014)
6. `http://automagic4android.com/en/help/base-concepts` (accessed on November 19, 2014)
7. `http://tasker.dinglisch.net/` (accessed on November 19, 2014)

1.2 Research Question and Scientific Methods

Within our research, we are going to answer the following questions:

Which contextual data is delivering relevant cues for the disruptiveness of interruptions on mobile devices? How can this knowledge be used to implement context–aware applications for alerts and reminders?

For finding appropriate contextual data, we will analyze the outcome of current research regarding this area and also implement a program, which is collecting contextual data and interrupting the user in regular time intervals. Based on this implementation we will conduct a study and try to find patterns within the study results gathered from our participants. For gathering implementation approaches, we will take a look at current literature dealing with the creation of context–aware implementations. These findings will be drawn in the context of implementing a reminder or alerting service for mobile devices.

As a final point in this introduction we will deliver an overview over the structure of this thesis, summing up the objectives of each chapter and their relation to the research questions.

1.3 Structure

This thesis is structured into four main areas, which are:

- *Interruption and Attention*, where we deliver theoretical background of the nature of interruptions and the human perception in connection to HCI. Within the discussion, involving research results and theoretical models, we find proof for the importance of context–awareness towards interruptions.
- *Context Aware Computing and Reminding*, where we evaluate the advantages of context–sensitive reminder systems, analyze and structure context data in more detail and finally discuss different approaches towards implementation suggestions. In the end, we compare our study to recent approaches with similar motivation. This chapter delivers a theoretical background for our final discussion regarding implementation approaches.
- *Study on Contextual Data*, where we describe the architecture of our study. The main point of this chapter is the results derived from our study, which are used for finding contextual clues in connection to the disruptiveness of interruptions.
- *Discussion* is finally bringing the findings of the previous chapters together for answering the research questions given.

2 Interruption and Attention

The research of interruptions has a long history, dating back to experiments in the 1920s (Gillie and Broadbent 1989). A relevant outcome for our specific study is that interrupting people affects their behavior (Daniel C. McFarlane 2002, 67). For instance, in early experiments Zeigarnik showed that interrupted tasks are more often recalled than uninterrupted ones due to people's selective memory relative to interruptions (Zeigarnik 1927). As in Zeigarnik's work also more recent research in interruption models focuses on demonstrating a relationship between interruption and memory (Latorella 1999, 5). For instance, Miller, Galanter and Pribram suggest, that the load of working memory explain heightened effects on recall and resumption of interrupted tasks (Miller, Galanter, and Pribram 1986). Although there are many studies referring to the motivational psychological background, there are less trying to address the degree of effect caused by interruptions (Latorella 1999, 5). But the research landscape started to change in the early years of this millennium as the influence of notifications in computer systems grew in importance (Cutrell, Czerwinski, and Horvitz 2001).

Serious research regarding the disruptive effects of interruptions started back in the late eighties with the research of Gillie and Broadbent, where they come to the conclusion that the nature and complexity of interruptions give direct information about their disruptiveness (Gillie and Broadbent 1989). Research regarding this area is important, as the disruptive effects of interruptions can have serious consequences. For instance, they can cause flight errors in commercial airline flights resulting in fatal crashes (Foushee and Helmreich 1988). In connection to interruptions on the flight deck, Latorella states that "several accidents are attributed to crews poorly integrating performance requirements for handling an interrupting system alert and compensatory actions with other aviation tasks" (Latorella 1999, 12). But also accidents in other complex systems, like for instance in power plants, can occur due to bad interruption management (Latorella 1999, 1). Therefore, Horvitz, Jacobs and Hovel take up the perspective that "human attention is the most valuable and scarcest commodity in human–computer interaction" (Horvitz, Jacobs, and Hovel 1999).

For our studies, we also focus on interruptions caused by HCI on human beings. In this context, Godbole and Smari define interruptions, which are system generated, as following:

> "A break in the continuity or uniformity of a user's activity, focus, or cognition, caused by system generated events during human–system interaction." (Godbole and Smari 2006)

Appropriate to this definition, McFarlane states, that one main disadvantage of complex computer systems, which are able to handle tasks without human control, is, that they need to interrupt the users away from the task they are performing whenever direct interaction is required (Daniel C. McFarlane 1997, 1). Unfortunately, the human perception only allows us to inspect the world with a limited spotlight of attention due

to limited cognitive abilities (Horvitz et al. 2003). In terms of economic resources this means that we have to deal with the effective usage of the limited time and attention of one individual (Hudson et al. 2002).

Also, Franke, Daniels and McFarlane remark, that this limited capacity affects the quality in decision–making in stressful situations and that, if this stream of information is not properly managed, these human capacities might get overloaded (Franke, Daniels, and McFarlane 2002). So there's a risk that the user might get disturbed in a non–desirable situation or time.

With focus on the research of HCI, Dabbish, Mark and González remark:

> "There is a sense that technology is fragmenting our attention by interrupting us (or enabling others to contact us), and so research in HCI has focused on understanding and preventing the negative consequences of external interruptions." (Dabbish, Mark, and González 2011)

Godbole and Smari sum up possible negative effects of such inappropriate interruptions as

- delays in terms of loss of time,
- manual or random errors,
- an increase of cognitive load when the user has to switch from an ongoing activity,
- loss of continuity when performing multiple tasks,
- loss of focus and concentration,
- socio–psychological effects as for instance annoyance, anger or frustration and
- the cost factors deriving from all these effects (Godbole and Smari 2006).

As it is crucial to avoid or absorb these effects as good as possible, we look into research about human perception generally and interruptions specifically and draw connections between our theoretical findings and our specific use case.

2.1 The Human Perception

To gather further data about the boundaries of human perception in the context of system generated interruptions, we look into theoretical models, which have been established over the past years and bring them in connection with our demands. We prove the limitations of human perception and also demonstrate the importance of context–awareness.

2.1.1 Model Human Processor

Card, Moran and Newell attempted to bring knowledge gathered from psychological literature into a model applicable for the design of interactive computer systems. In designing such a model they screened the abilities to do "task analysis (determining the specific, rational means of accomplishing various goals), calculation (zero–parameter predictions of behavior capable of parametric variation) and approximation (simplification of the task and psychological theory)" as important to succeed. Therefore, they introduced the Model Human Processor, which has been described as following:

> "The Model Human Processor can be described by (1) a set of processors, memories and their interconnections together with (2) a set of "principles of operation". The principal properties of the processors and memories are summarized by a small set of parameters." (Card, Moran, and Allen Newell 1986)

The basic principle is, that human cognition is depicted by three parallel processors, which are the Perceptual Processor for sensory information, the Cognitive Processor for information processing and the Motor Processor controlling external actions as outlined in Figure 1. Card, Moran and Newell remark, that the cognitive system is acting parallel in recognition, but serial in its action phase, allowing doing only one thing at a time (Card, Moran, and Allen Newell 1986).

This model has for instance been used for measuring the usability of mobile devices by comparing model predictions within samples of different user groups (Jastrzembski and Charness 2007). For our research, this example proves again the limits of human perception, where a queue of actions can only be executed and perceived sequentially at a specific point of time. McFarlane's and Latorella's formulation that people "think in parallel and act in serial" also confirms this assumption (McFarlane and Latorella 2002, 4).

2.1.2 GOMS Models

Card et al. applied some of their ideas of the Model Human Processor to establish analysis tools called GOMS (Goals, Operators, Methods and Selection rules) Models (Daniel C. McFarlane 1997, 12). Since its introduction, this model has become one of the most widely known theoretically concepts in HCI (John and Kieras 1996, 3). McFarlane describes it as following:

> "GOMS can be employed to model how a person would perform a given task. The task is analyzed hierarchically into the subtasks that comprise it ("Goals"). These subtasks are modeled with chains ("Methods") of basic operations ("Operators") that must be performed to accomplish them. The "Selection rules" are productions to simulate which chain a person would choose among alternatives to complete a subtask depending on the context." (Daniel C. McFarlane 1997, 12)

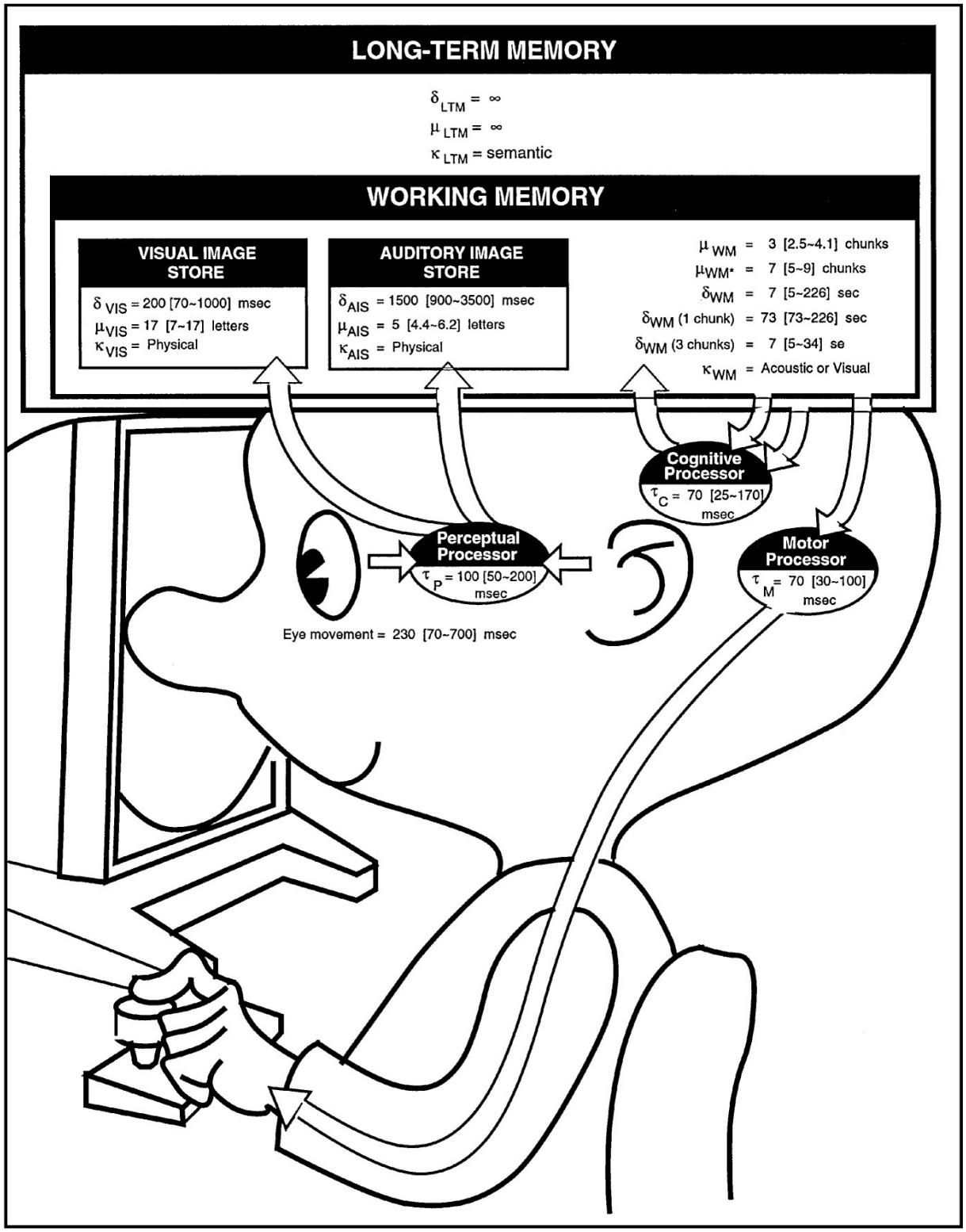


Figure 1: The Model Human Processor can be described as following: "Sensory information flows into Working Memory through the Perceptual Processor; Motor programs are set in motion through activation of chunks in Working Memory. Working Memory consists of activated chunks in Long Term Memory. The Visual Image Store and Auditory Image Store can be thought of as special activations of the experiental and analogic properties of visual and auditory images." (Card, Moran, and Allen Newell 1986).

Adamczyk, Iqbal and Bailey implemented a method for monitoring user tasks and reasoning about interruptions using the GOMS Models. They argument, that breakpoints in task models are important key moments for interruption. To find those moments, they bring up a connection between the pupillary size and the mental workload as they state — based on previous research (Cellier and Eyrolle 1992)— that the best moment of interruption is "between two coarse breakpoints that are, on the whole, better understood and better recalled than other points in the task" (Adamczyk, Iqbal, and Bailey 2005).

In some other research, Adamczyk and Bailey introduce a task model, where they additionally split up these coarse breakpoints into fine breakpoints, which "can be understood as subtasks of a larger coarse breakpoint" (Adamczyk and Bailey 2004). They predict the best points of interruptions as "those when a user is moving from one well–defined and commonly understood task to another, not simply between any two subtasks" (Adamczyk and Bailey 2004). They cause interruptions by triggers, which they explain as following:

> "Interruption triggers are based on behavior believed to be significant in the mind of the user, and the interruptions are not associated with a temporal phase, making it easier for them to be applied anywhere during execution." (Adamczyk and Bailey 2004)

The basic setup for their task model hierarchy is outlined in Figure 2. For the setup of a mobile device, we try to achieve finding those desirable breakpoints by drawing conclusions to the user's context using sensory data as outlined in Chapter 4.2.1.

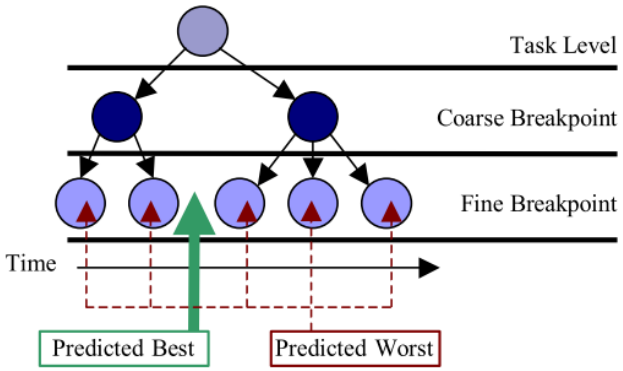


Figure 2: This image illustrates the task model of Adamczyk and Bailey where they predict the moment between two fine breakpoints as the best moment of interruption (Adamczyk and Bailey 2004).

As the Model Human Processor, the original GOMS model doesn't support the modeling of multitasking (Daniel C. McFarlane 1997, 13), which is important in terms of interruptions. John and Gray extended this model for multitasking in the CPM–GOMS (Critical Path Method–GOMS) by employing the idea of three separate processors as in the Model Human Processor (John and Gray 1995). In short, this leads to an implementation of

fundamental aspects of the structure of human cognition, meaning that "people can do things while they wait for themselves to finish doing other things" (Daniel C. McFarlane 1997, 13).

With the knowledge of human multitasking capabilities and the prior mentioned limits of human perception we come to the finding, that people might be overwhelmed by too many complex parallel tasks and are therefore vulnerable to distraction. Preece et al. sum up the danger of distraction as following:

> "While most people show great flexibility in coping with multitasking, they are also prone to distraction. On returning to a suspended activity, it is possible for them to have forgotten where they were in the activity. As a result they may not restart from where they left off but will recommence at a different point of entry." (Preece et al. 1994, 105)

In connection to our research, we should examine how often and for how long the user is using his or her mobile device for being aware whether or how much the user will be distracted during an interruption. This proves the relevance of context–awareness when implementing a system causing interruptions. Therefore, we discuss this topic on a bigger scope in Chapter 2.2.3.

2.1.3 Keystroke–Level Model

The Keystroke–Level Model measures "the time it takes a user to perform a task with a given computer system" (Card, Moran, and Allen Newell 1980). Holleis et al. altered this model for interactions on mobile phones (Holleis et al. 2007). Though mainly used for improving interaction performance in terms of HCI, we will describe this model for helping us to understand how a user interacts with a mobile device and establish a connection to possible interruptions.

Holleis et al. introduce a new operator called "Macro Attention Shift" modelling the time needed to "shift the focus between the contents on the screen of the mobile device to an object (e.g., a poster) in the real world and vice versa" (Holleis et al. 2007). Also, a new relevant operator is "Distraction", dealing with distraction time values from the real world while handling a mobile device. To acquire time values, they recruited a user group performing a task within a test scenario. Of interest for us is their finding for evaluating distraction:

> "People cope with such situations in different ways. They use their peripheral view, make quick glances, or introduce pauses. Initial tests showed that the behavior also depends on the type of task. Thus, distraction cannot be easily modelled as certain specific actions. Through our tests we found that it is more appropriate to model distraction as a multiplicative factor rather than an additive operator." (Holleis et al. 2007)

Additionally, they made it clear that complex interaction styles are handled in a different way by different user groups. For our research, we adopt the finding, that distraction itself has an impact on all active tasks and shouldn't be modelled as independent action. So for a reminding system, an alert influences the whole context, which is depending on the user him– or herself.

2.1.4 The Resource Competition Framework

Oulasvirta et al. introduce the Resource Competition Framework (RCF) as an attempt to describe the cognitive faculties for controlled selection and division of attention in the sector of mobile HCI. RCF is based upon three cognitively oriented traditions. First, RCF builds up on the tradition of task analysis, by analyzing the tasks a person is performing in a situation and "decompose them to their components, and identify related mental requirements" (Oulasvirta et al. 2005). Also, the RCF aims to "identify and explicate the relevant cognitive resources and their properties" and it uses knowledge from cognitive psychology and tailors it to a mobile use case (Oulasvirta et al. 2005).

They conducted a study where participants had to follow a route and fulfill different browser tasks on a mobile device (see Figure 3). While doing so, their behavior, action and context were recorded. Their results give us some insights into the short attention spans when performing mobile tasks while also keeping up awareness of the environment:

> "Our results demonstrate that resource competition is very real and seriously constrains mobile interaction. The data conveys the impulsive, fragmented, and drastically short–term nature of attention in mobility. Continuous attention to the mobile device fragmented and broke down to bursts of just 4 to 8 seconds, and attention to the mobile device had to be interrupted by glancing the environment up to 8 times during a subtask of waiting a Web page to be loaded." (Oulasvirta et al. 2005)

Additionally, they come to the outcome, that the aspects of managing a mobile context are psychosocially more important to an individual than HCI tasks on the mobile device. For instance, it was not easy for the participants to spend more awareness to a task on the phone (by telling them that they should hurry up) and lower the attention to the environment. On the other hand, they easily pushed the HCI task down when they were told to spend more awareness to the surroundings (Oulasvirta et al. 2005).

For our specific use case, the studies of Oulasvirta et al. deliver evidence for the sparse attention resources within mobile environments, as people in general tend to be more distracted by the environment due to the mobile nature of these devices. Also it appeals to be very important to take the context of the people in account, as there are always attentional resources spent into managing the surrounding context. As context–awareness is a very important point for our studies, we will discuss its relevance and general content in more detail in Chapter 3.

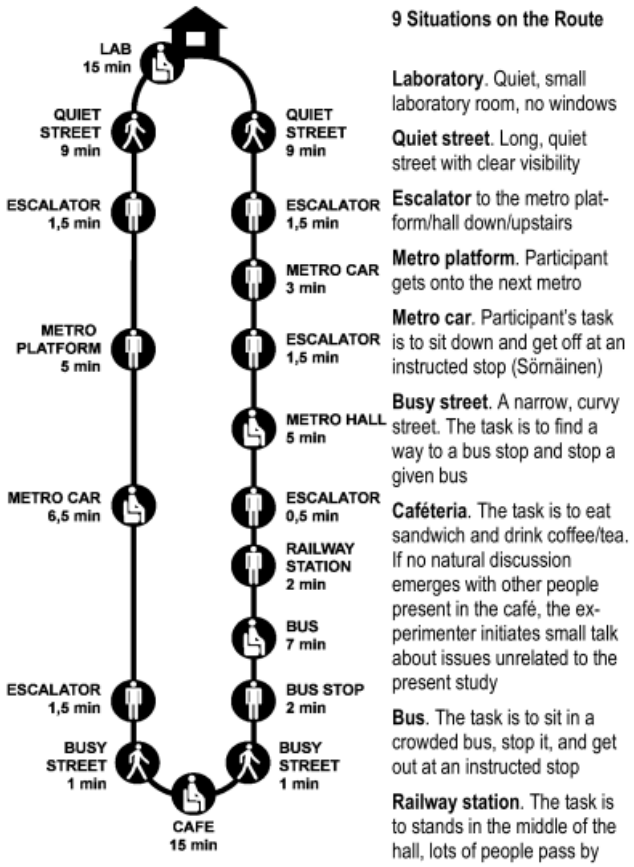


Figure 3: Oulasvirta et al. conducted a study to describe the selection and division of attention in the sector of mobile HCI. The participants had to follow a route while fulfilling different browser tasks on their mobile device. The route along with their real–world tasks is outlined in this image (Oulasvirta et al. 2005).

2.2 Interruptions in HCI

In this chapter, we analyze the way how people deal with interruptions by discussing the Interruption Stage Model. Then, we identify the problems of human interruption and draw connections to our specific use case. The problem of task switching will be tackled in more detail and solutions for dealing with the loss of continuity will be presented. Next, we discuss about strategies for deciding about the right moment to interrupt and find proof for the importance of the user's context. Finally, we draw the important findings of this chapter in the context of mobile applications.

2.2.1 Interruption Management Stage Model

The IMSM (Interruption Management Stage Model) is "the first thorough model–based treatment of how people deal with interruptions" (McFarlane and Latorella 2002, 17). To

formalize the process of interruption management, Latorella defines four different stages, which are:

- The detection stage, where the interruption is recognized and stored in short–term memory,
- the interpretation stage, where the memorized interruption is translated and interpreted,
- the integration stage, including "sub–stages of ongoing task preemption, interruption performance/scheduling, and ongoing task resumption" and finally
- termination and continuation with the ongoing task (Latorella 1999, 21).

Also, she defines the effects, occurring when finalizing these stages. In the beginning, diversion draws the user's attention away from his or her current task, distracting and disturbing when the interruption is integrated. Also future performance may be disrupted on the ongoing procedure (Latorella 1999, 23). The execution sequence of these stages including the previous mentioned effects is represented in Figure 4.

McFarlane and Latorella summarize the usage of this model as following:

> "This model serves to (a) organize basic research addressing perception, memory, attention, motivation, scheduling, and planning to identify task (interrupted and interrupting), environment, and operator factors relevant to interruption management; (b) characterize interruption management as information processing stages — with the understanding that it is a simplification of actual mental processes; (c) characterize people's interruption management behaviors in the context of these stages; (d) characterize the deleterious effects of interruptions in terms of these stages; and (e) suggest dependent measures useful for sensitively measuring these deleterious effects." (McFarlane and Latorella 2002, 15)

To sum up these ideas for our specific usage, we use the ideas behind interruption management to identify basic research in the field of psychology and alter and simplify the outcomes for our needs. Of importance for us is the knowledge about how and in which ways a user responses to an occurring interruption.

Latorella distinguishes between five possible behaviors, which are "detection and oblivious dismissal", "interpretation and unintentional dismissal", "integration and intentional dismissal", "integration and preemptive integration" and finally "integration and intentional integration" (Latorella 1999, 117–124).

McFarlane and Latorella describe and enumerate those reactions as following:

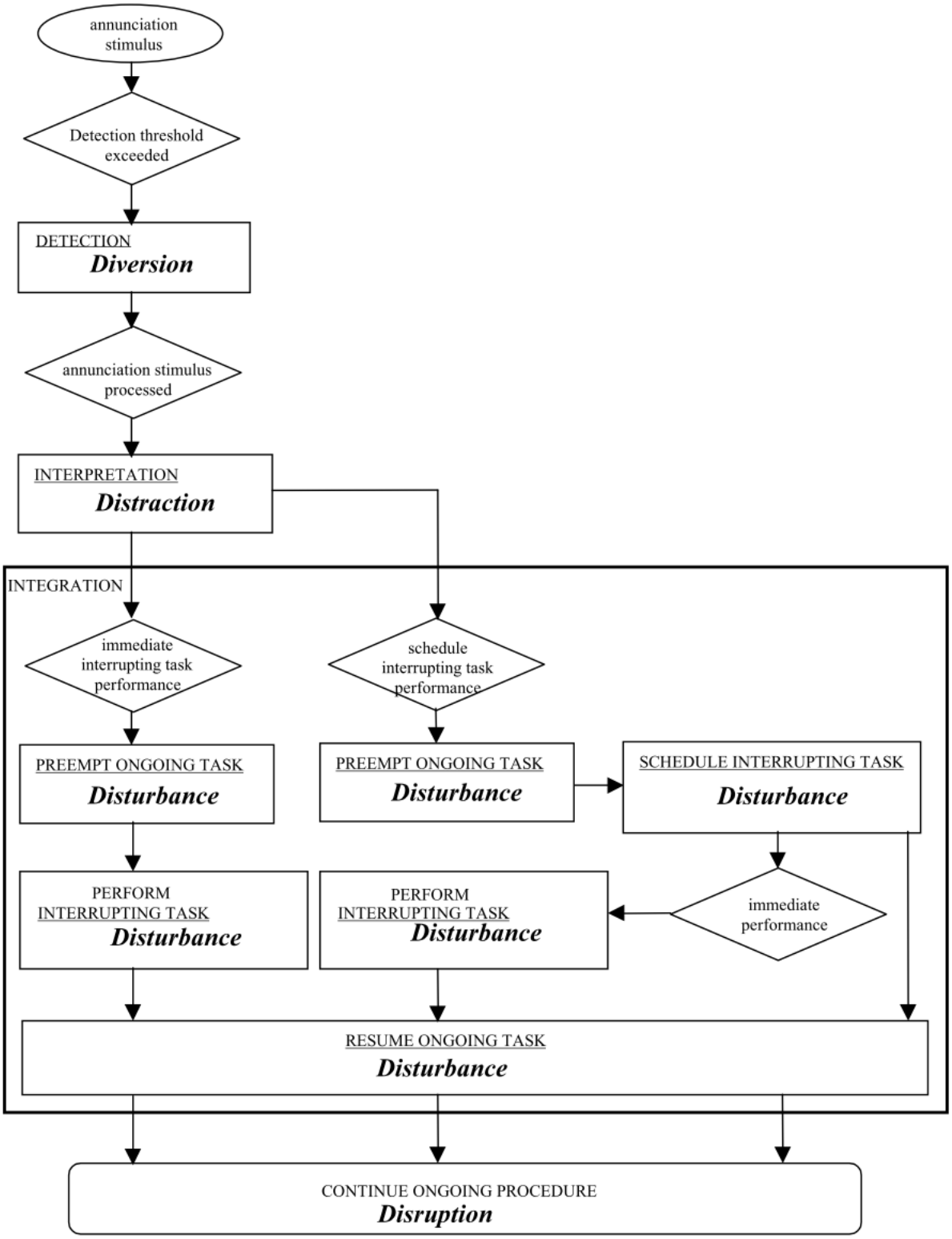


Figure 4: This figure outlines the course of actions in the Interruption Management Stage Model in more detail and the correlation between diversion, distraction, disturbance and disruption (Latorella 1999, 25).

> "(a) oblivious dismissal, the interruption annunciation is undetected and the interruption is not performed; (b) unintentional dismissal, the significance of the annunciation is not interpreted and the interruption is not performed; (c) intentional dismissal, the significance is interpreted, but the operator decides not to perform the interrupting taks; (d) preemptive integration, the interrupting task is initiated immediately, intruding on the ongoing task, and performed to completion before resuming the ongoing task; and (e) intentional integration, the interrupting task and the ongoing task are considered as a set, and the operator rationally determines how to integrate performance of the interrupting task." (McFarlane and Latorella 2002, 17)

The separation between the different reactions in the last paragraph is something we need to consider when designing a system for creating context–sensitive alarms. Especially the sparse diversion between dismissal and integration is something we need to concentrate on. In Chapter 4.2 we outline our approach in our implementation bearing these theoretical findings in mind.

2.2.2 Taxonomy of Human Interruption

McFarlane identifies eight dimensions of the problem of human interruption:

- The first factor is the source of interruption (McFarlane and Latorella 2002, 18–19). It can be versatile (see Figure 5), but, as mentioned in Section 2.2, we concentrate on interruptions caused by HCI on humans; so in our use case the interruption source is a (mobile) computer.
- The next factor is a definition of the person's characteristic receiving the interruption. McFarlane for instance distinguishes the abilities to multitask, the level of anxiety, the ability to maintain a constant level of arousal or the degree of coordination (McFarlane and Latorella 2002, 20).
- The method of interruption distinguishes whether the interruption occurs immediate, negotiated, mediated or scheduled (McFarlane and Latorella 2002, 25). In our test implementation, we generate only scheduled interruptions as we want to collect data about the user's context.
- The meaning of interruption describes the reason why the user gets interrupted (McFarlane and Latorella 2002, 20). Our reason for interruption consists only of measuring the disturbance of the interruption itself to bring it in connection to the sensor data of the device.
- The method of expression is about how the interruption gets expressed to the user. One main goal of this method is to mitigate the negative effects on user performance (McFarlane and Latorella 2002, 21). In our test setup, we always show the same message box to apply an interface someone gets used to and to give the ability to respond fast.

- The channel of conveyance gives information whether the interruption is delivered by a direct communication channel, mediated by a person, a machine or some other animate object (McFarlane and Latorella 2002, 21–22). As our research goal is to gather data about the user's context within a mobile device, our channel of conveyance is the delivery by a machine.
- The human activity changed by interruption covers the aspects on how the people's performance changes after an interruption, measuring how disruptive the interruption is in the user's context (McFarlane and Latorella 2002, 22–23).
- The final factor is the effect of the interruption, evaluating whether the interruption had a negative effect and reasoning about the occurrence of the effect (McFarlane and Latorella 2002, 23).

While in our implementation some of these effects are very easy to categorize (such as the source or meaning of interruption), we attempt to regenerate other parameters (like the effect of the interruption) by analyzing user data in connection to the disturbance. Recapitulating, all of these factors offer us a compromising overview over the points that need to be considered when creating software causing interruptions.

2.2.3 Task Switching

In Chapter 2.1 we already discussed the limitations of human cognition. Based on this knowledge, we outline more specifically the challenges occurring, when switching between the current and the interruptive task during receiving an interruption in the user's workflow. Within medical background, Coiera discusses the importance of task analysis as following:

> "Understanding that the impact of an interruption is dependent on its position in a sequence of tasks, how the interruption is handled or by the existence of memory cues in the working environment to assist task resumption after interruption, all help us explain the complex phenomena we observe in clinical settings." (Coiera 2012)

Czerwinski, Horvitz and Wilhite conducted a diary study where they categorized the amount of task switching and the successive interruptions experienced by knowledge workers over the course of a working week. They come to the following conclusion in evaluating the difficulty of switching:

> "The set of results shows that task complexity, task duration, length of absence, number of interruptions, and task type influence the perceived difficulty of switching back to tasks. Specifically, complex, "returned–to" tasks comprise a significant portion of an information worker's week, but reacquiring such tasks is considered difficult by users." (Czerwinski, Horvitz, and Wilhite 2004)

Factor of Human Interruption	Example Values
Source of interruption	Self [human], another person, computer, other animate object, inanimate object.
Individual characteristic of person receiving interruption	State and limitations of personal resource (perceptual, cognitive, and motor processors; memories; focus of consciousness; and processing streams); sex; goals (personal, public, joint); state of satisfaction of face-wants; context relative to source of interruption (common ground, activity roles, willingness to be interrupted, and ability to be interrupted).
Method of coordination	Immediate interruption (no coordination); negotiated interruption; mediated interruption; scheduled interruption (by explicit agreement for a one-time interruption, or by convention for a recurring interruption event).
Meaning of interruption	Alert, stop, distribute attention, regulate dialogue (meta-dialogue), supervise agent, propose entry or exit of a joint activity, remind, communicate information (illocution), attack, no meaning (accident).
Method of expression	Physical expression (verbal, paralinguistic, kinesic), expression for effect on face-wants (politeness),[a] signaling type (by purpose, availability, and effort), metal-level expressions to guide the process, adaptive expression of chains of basic operators, intermixed expression, expression to afford control.
Channel of conveyance	Face-to-face, other direct communication channel, mediated by a person, mediated by a machine, meditated by other animate object.
Human activity changed by interruption	Internal or external, conscious or subconscious, asynchronous parallelism, individual activities, joint activities (between various kinds of human and non-human participants), facilitation activities (language use, meta-activities, use of mediators).
Effect of interruption	Change in human activity (worth of this change is relative to the person's goals), change in the salience of memories, change in awareness (meta-information) about activity, change in focus of attention, loss of willful control over activity, change in social relationships, transition between stages of a joint activity.

Figure 5: Here, the factors of human interruption — defined by McFarlane — are outlined with corresponding example values (McFarlane and Latorella 2002, 19).

O'Conaill and Frohlich also conducted a study observing the nature of interruptions in the workspace, where they state, that "many interruptions (41 percent) result in the discontinuing of the interrupted task beyond the duration of the interruption itself" (O'Conaill and Frohlich 1995). Also, Franke, Daniels and McFarlane state about task switching, that the "cognitive demands of these context switches increase the effective workload of users, which in turn increases the probability of mental mistakes" (Franke, Daniels, and McFarlane 2002). On the other hand, a focus on "attention is critical to solving problems or completing complex tasks that require a great deal of information to be held in working memory" (Dabbish, Mark, and González 2011). We'll discuss in Chapter 2.2.4 that interruptions — following task switching — cannot always be avoided.

For being more specific, McFarlane and Latorella introduce interruption phases under the point of view, that a human person must switch his or her current task to the interruption task and vice versa. They explain three phases, which are before, during and after the switch (McFarlane and Latorella 2002, 46).

The main motivation for the before switch phase is to ensure that people are interrupted in the best possible way. To achieve this, the interruption should be predictable, with an UI supporting this attribute switch (McFarlane and Latorella 2002, 46–47). It should allow the user to rehearse the current task by some cue such as "a visual flash, an audible beep, or a vibration" (Franke, Daniels, and McFarlane 2002).

In the during switch phase the emphasis is on maximizing both the performance on the interrupting and the inherent task, focusing whether the task is completed or create awareness if not. Finally, the after switch phase is responsible for minimizing the disruption of the prior interruption by supporting resumption of the interrupted tasks (McFarlane and Latorella 2002, 47). Franke, Daniels and McFarlane support this context–recovery by "providing the user commands that query the interface about aspects of the previous task" like for example in the form of a meta–dialogue asking questions like "What was I last working on?" (Franke, Daniels, and McFarlane 2002). Cutrell, Czerwinski and Horvitz also see potential in the idea of reminding the user of his or her prior task and assume that this might be helpful for getting back to working routine more quickly after an interruption (Cutrell, Czerwinski, and Horvitz 2001).

Altmann and Trafton measured the resumption lag, which is the time needed to restart a task after an interruption is over. Their observations on the overall time course of an interruption is outlined in Figure 6. In their user studies they come to the conclusion, that the resumption lag is double the interval between uninterrupted actions. They also examined the role of external cues, "finding that cues available immediately before an interruption facilitate performance immediately afterwards (reducing the resumption lag)" (Altmann and Trafton 2004). They find an explanation by the observation that "people deploy preparatory perceptual and memory processes, apparently spontaneously, to mitigate the disruptive effects of task interruption" (Altmann and Trafton 2004).

One statement, that can be adopted for our use case, is the main cause for optimizing the before switch phase by making interruptions predictable. In the context of alerts it appeals rewarding by letting the user identify patterns within his or her own context when an interruption occurs. On the other hand, our implementation doesn't take the context of the user in account, as we want to collect data about distinguishing between appropriate and not appropriate interruptions. Nevertheless, we achieve a simple form of predictability by evoking interruptions at regular time intervals. As for cues on interruption, we use the alert sound selected by the user for notifications, supported by the phone's capability to vibrate.

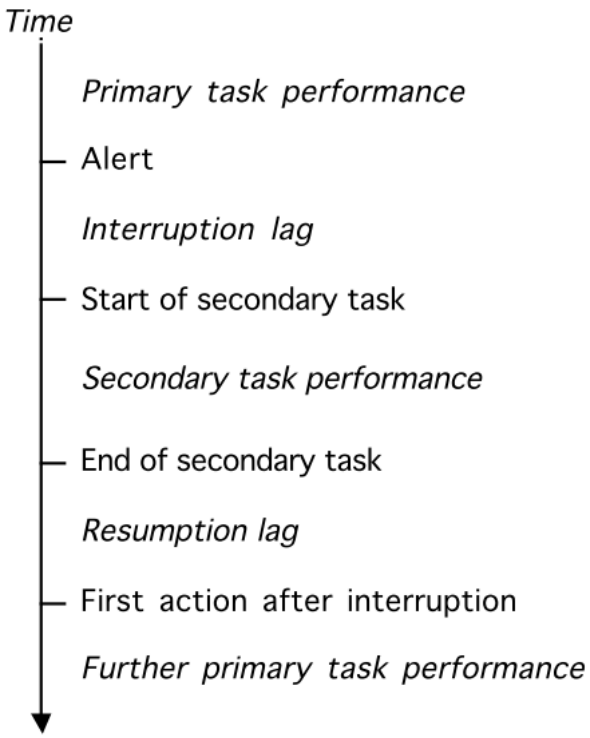


Figure 6: In this figure, the time course of an interruption regarding task switching is illustrated (Altmann and Trafton 2004).

2.2.4 Measuring and Predicting Interruption Time

Harr and Kaptelinin state, that a natural strategy in avoiding the disruptive effects of interruptions is by simply stop them from happening. Unfortunately, this approach is very difficult to implement in computer–mediated communication, where people want to be constantly available, yet not to all sorts of communication (Harr and Kaptelinin 2007). This issue can be addressed via availability management, where the user gives direct information about his or her availability or presence to manage whether interruptions are appropriate or not. This can for instance be achieved by setting the availability of a messenger application to "occupied" or "away" to suppress notifications about incoming messages (Harr and Wiberg 2008, 244). For our specific use case, we assume that the user doesn't want to get interrupted by alerts while he or she is sleeping. So we offer the possibility to set up a resting time, where no alarms will occur (see Chapter 4.1). Of course that's not the ultimate solution in managing interruptions as "most people prefer taking the trouble of dealing with interruptions to shutting down completely from external interruptions and thus running the risk of missing valuable information" (Harr and Kaptelinin 2007). Therefore, it is appropriate to find the moment, where interruptions are less disruptive for the user.

Adamczyk and Bailey conducted an evaluation to measure the effects of interruption at different point of times. They chose certain moments of task execution in terms of

task performance, emotional state and social attribution. They tried to predict the best moment of an interruption using task model hierarchy by splitting up the current task into breakpoints as discussed in Chapter 2.1.2. For gathering data, they let a variety of subjects perform office tasks on the computer while causing interruptions. They come to the conclusion, that the interruption timing has a significant main effect on the reported annoyance and frustration. The mental effort and demand is in direct connection with the task type itself. As a result, their predicted best moments were effective at reducing the disruptive effects of the interruption by producing "less annoyance, frustration, and time pressure" and requiring less mental effort (Adamczyk and Bailey 2004).

As their tasks and also their predictions of disturbance within these tasks were predefined, we cannot alter their approach for our use case, where the tasks performed by users are much more versatile, but nevertheless their research is one more argument for the importance of evaluating the best possible interruption time.

Franke, Daniels and McFarlane chose a strategy where they decided about the interruption time on a case–by–case basis, which they outline as following:

> "Our selection criteria is based on a dynamic automated assessment of the relative importance between the current task and the interrupting task. If the interrupting task is mission critical compared to the current task, the user is interrupted immediately. If the current task is critically important compared to the interrupting task, the alert is held until the user is finished with the current task (that is, it's scheduled for the next cognitive break). In all other cases, the interruption is negotiated." (Franke, Daniels, and McFarlane 2002)

For achieving that, they prioritize both the interruption and the current task with low, medium or high priority and select an interruption strategy based on these assumptions (Figure 7). For our implementation, this approach appeals promising, yet the prioritization of the importance of the current task in connection to the user's context is something we want to achieve by analyzing the acquired user data (see Chapter 4.3).

Interruption Task	Current Task		
	high	medium	low
high	Negotiated, default defer	Negotiated, default interrupt	Interrupt immediately
medium	Negotiated, default defer	Negotiated, default defer	Negotiated, default interrupt
low	Defer interruption until cognitive break	Negotiated, default defer	Negotiated, default defer

Figure 7: Franke, Daniels and McFarlane created a table chosing the interruption strategy depending on the current task and the interruption task (Franke, Daniels, and McFarlane 2002).

2.2.5 Interruptions within Mobile Applications

In the past chapters, we already brought up our findings about interruptions and the human perception in the context of mobile computing. Here, we'll discuss studies dealing with interruptions in the context of mobile applications specifically, and prove, that our prior outcomes can be adopted for a mobile use case.

As task interruption is an inherent problem in smartphones, Leiva et al. conducted a study to gather data about intended and unintended interruptions on mobile phones (Leiva et al. 2012). They looked into the cost of interruptions caused by task switching and incoming calls in terms of task completion time and come to the conclusion, that such interruptions rarely happen, but introduce a significant overhead when they do (Leiva et al. 2012). Böhmer et al. try to tackle this problem by implementing a new UI that doesn't switch to full screen when a call arrives, giving the users the opportunity to stay in their current task till they are ready for a task switch (Böhmer et al. 2014). This implementation follows the suggestion of Leiva et al. to help the user to remind the context while switching to another application. They also suggest to return to the interrupted application once the interruption has been handled (Leiva et al. 2012). We already outlined several studies giving comparable recommendations in the issue of task recovery and context retrieval (see Chapter 2.2.3).

Ho and Intille used accelerometers for activity recognition to identify appropriate opportune moments for mobile interruptions. In their studies, they showed that "proactive messages delivered by a mobile computing when the user is transitioning between two physical activities (e.g. sitting to walking) may be received more positively than the same messages delivered at random times" (Ho and Intille 2005). In the discussion about GOMS Models in Chapter 2.1.2 we also come to similar results, where the best moment for interruptions is identified between two coarse breakpoints (Adamczyk, Iqbal, and Bailey 2005).

Poppinga, Heuten and Boll triggered notifications within a study trying to find the right moment for interruptions within the user's context. They come to the conclusion, that the right time and whether the user holds the phone in his or her hands or not are critical factors for the disruptiveness of an interruption (Poppinga, Heuten, and Boll 2014). As their overall research follows our approach very closely, their research will be examined in more detail in Chapter 3.4. Also Pielot, Church and Oliveira imply, that the amount of how busy users are with other tasks is in direct connection to the time when interruptions are dealt with (Pielot, Church, and Oliveira 2014). Additionally, Ferreira et al. indicate as a result of their studies that "people are more open towards interruptions at home, compared to the university and work" (Ferreira et al. 2014). These results demonstrate once more the importance of context–awareness in connection to measuring the disturbance of interruptions.

In conclusion, the disruptive nature of interruptions on smartphones — measured for instance by Leiva et al. in terms of incoming calls — lead to a lot of potential for future interface and interruption management strategies on mobile devices. We'll discuss the current approaches for actual implementations in this field briefly in Chapter 3.4.

2.3 Synopsis

In this chapter, we discussed general and more recent approaches in research of human attention with focus on perception and interruptions in terms of HCI. We delivered a basic overview over the transformation of the research landscape from interruptions and memory to the investigation of their disruptive effects caused by the influence of computer systems. A definition of interruptions in terms of HCI was given and possible negative effects of inappropriate interruptions were listed.

The limitations of the human perception were further analyzed by discussing popular models used in HCI research. We found proof within the Model Human Processor which illustrates that people "think in parallel and act in serial" (McFarlane and Latorella 2002, 4). Also, the GOMS Models gave us insight in how people perform tasks and we looked into extensions of this model, illustrating the multitasking behavior of human beings and dealt with the problem of distraction within complex tasks. We defined the impact of distractions in more detail, using findings of the Keystroke–Level Model and came to the conclusion, that distraction is not an independent action as it has impact on all active tasks. For our discussion about task execution, we looked into the Resource Competition Framework and drew a line to context–awareness within mobile devices using results of a study where the distractive effects of the user's surroundings were measured (Oulasvirta et al. 2005).

For interruptions in the context of HCI, we first had a look into the Interruption Management Stage Model, formalizing the process of interruption management and describing possible reactions on machine driven interruptions. The problems of human interruptions were defined and connections to our specific use case were drawn. We tackled the problem of task switching in more detail and observed, that it is important to support the resumption of interrupted tasks by context–recovery, supporting the user reminding his or her prior task before interruption. Also for our use case we came to the conclusion, that it appeals rewarding by designing interruptions predictable, by for instance firing alarms at regular time intervals. The importance of predicting the right interruption time was mentioned by not letting interruptions occur at unwanted time intervals (like sleeping times) and decide about appropriate moments for interruption. For finding these moments, the task type and its complexity and importance should be taken in account as well if a task or parts of it have been finished by splitting them up into breakpoints. Finally, we brought the important points of these results in direct connection to the context of mobile applications and found proof in recent studies, that our findings in interruption management are amenable for the field of mobile computing.

Multiple times we found arguments for the importance of context–awareness in deciding about the right time of an interruption. Therefore, we will take a closer look into the field of context aware computing in the next chapter.

3 Context Aware Computing and Reminding

In Chapter 2.2 we already pointed out the importance of the user's context in terms of interruptions. In opposite, also the effects of interruptions are context–specific (Coiera 2012). Schilit, Adams and Want define three important aspects of context as "where you are, who you are with and what resources are nearby". Additionally, they remark:

> "Context encompasses more than just the user's location, because other things of interest are also mobile and changing. Context includes lighting, noise level, network connectivity, communication costs, communication bandwith, and even the social situation; e.g. whether you are with your manager or with a co–worker." (Schilit, Adams, and Want 1994)

Within our studies in this thesis we concentrate on the creation of a context–aware system, which Baldauf, Dustbar and Rosenberg define and describe in the context of mobile applications as following:

> "Context–aware systems are able to adapt their operations to the current context without explicit user intervention and thus aim at increasing usability and effectiveness by taking environmental context into account. Particularly when it comes to using mobile devices, it is desirable that programs and services react specifically to their current location, time and other environment attributes and adapt their behaviour according to the changing circumstances as context data may change rapidly. The needed context information may be retrieved in a variety of ways, such as applying sensors, network information, device status, browsing user profiles and using other sources." (Baldauf, Dustdar, and Rosenberg 2007)

Research about context–aware systems is dated back to the early 1990s by the introduction of the Active Badge Location System, which is considered to be one of the first context–aware applications (Baldauf, Dustdar, and Rosenberg 2007). It used infrared technology to identify the user's location in order to forward phone calls to a telephone nearby (Want et al. 1992). In sum (and also implied by the definition of Baldauf, Dustbar and Rosenberg above), the area of context–aware computing is about increasing the relevance of the user's context towards the creation of applications and their content (Chang 2013). Over the time, the popularity of this area has increased significantly and been established as well–known research area in computer science. Also, sensors for gathering data got more powerful, cheaper and smaller (Perera et al. 2014). Both the technological improvements and the increase of awareness within research justify the importance of context–awareness for our research. As an overall result, research in context–aware computing is targeted to bring us one step closer to the vision of pervasive computing by enabling computer systems to anticipate the user's needs and act in advance (Chen and Kotz 2000). In more recent literature, context–aware (pervasive) computing is also identified as the most recent

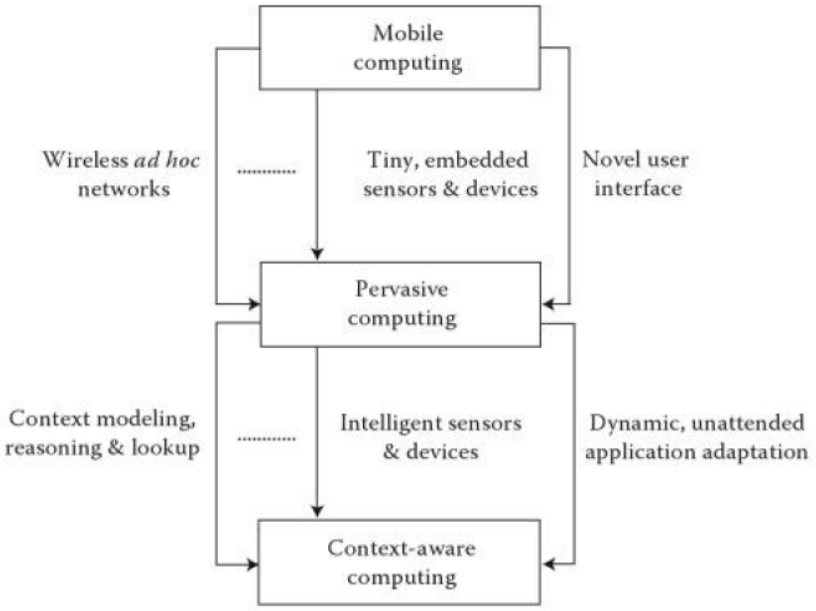


Figure 8: This figure outlines the connection between mobile computing, pervasive computing and context–aware computing (Kumar and Xie 2012, 271).

area of research within the evolution from mobile to pervasive computing as illustrated in Figure 8 (Kumar and Xie 2012, 271–272).

Especially within the area of mobile devices, context–aware computing is both considered as necessary and feasible. The necessity is derived from the small mobile phone display, where "information must be delivered with much higher relevance and precision to meet user needs" (Chang 2013); the feasibility is justified "because small, light–weight mobile devices allow users to almost always carry them around, and much can be learned via a phone about its user's habits and states" (Chang 2013). As a result, lots of research has been done to support minimizing or nullifying the disrupting effects of interruptions (outlined in more detail in Chapter 2.2) on mobile devices (Zulkernain, Madiraju, and Sheikh Iqbal Ahamed 2011).

Generally speaking, one main goal of context–aware computing is to enable computer systems to anticipate user's needs and act in advance, to free them from manual instruction and configuration (Chen and Kotz 2000). Schilit, Adams and Want categorize these approaches into the following categories:

- Proximate selection, where nearby objects are favored. These objects can be computer input and output devices, non–physical objects and services (like bank accounts or list of instructions) or the set of places someone wants to find out about.
- Automatic contextual reconfiguration, where components are removed, added or their connections are altered due to context changes.

- Contextual information and commands, where "queries on contextual information can produce different results according to the context in which they are issued" (Schilit, Adams, and Want 1994).
- Context–triggered actions, where IF–THEN rules are specified to describing how context–aware systems should adapt.

These categories are assigned to context–aware software dimensions telling "whether the task at hand is getting information or carrying out a command and whether it is effected manually or automatically" (Schilit, Adams, and Want 1994). These dimensions are also outlined in Figure 9. When taking our approach into account, we're interested in actions, which are taken automatically for both information and command. As a result, automatic contextual reconfiguration and context–triggered actions are of interest for us. Those coarse subdivisions will be discussed in greater detail within topics like rule based reasoning or machine learning reasoning in Chapter 3.3 of this section.

	manual	automatic
information	proximate selection & contextual information	automatic contextual reconfiguration
command	contextual commands	context-triggered actions

Figure 9: In this table, context–aware software dimensions are assigned, depending if the task at hand is an information or command and if it is effected manually or automatically (Schilit, Adams, and Want 1994).

As we're focusing on using this knowledge to apply to a reminder system, we will discuss prior research in context–sensitive reminding in more detail in the following chapter and draw the conclusions within this section in context to reminder systems.

3.1 Reminder Systems

Reminder systems deliver assistance by generating awareness upon upcoming events, recalling tasks or delivering additional information, like for instance the names of people attending a meeting. This can lead to enhanced task efficiency and outcomes (Kamar and Eric Horvitz 2010, 2). In the past, several of such systems have been proposed. For instance, around the millennium comMotion has been introduced as a location–aware computing environment, allowing to set up reminders based on the user's location (Marmasse and Schmandt 2000). Also systems like Cyreminder, The Forget–Me–Not-System or the Towel System pursue the implementation of context–sensitive reminding (Kamar and Eric Horvitz 2010, 1). But also more general approaches, such as to–do lists written on a piece of paper in form of e–mails or on post–it notes and the support of personal information management tools (such as electronic calendars) or human assistants have been suggested (Dey and Abowd 2000).

Kamar and Horvitz describe an ideal reminder system as following:

> "An ideal reminder system should consider both the benefit that a reminder may generate for the user and the cost of interruption associated with transmitting the reminder." (Kamar and Eric Horvitz 2010, 2)

Kamar and Horvitz further draw conclusions between the benefits and the cognitive state of a user:

> "The benefit of a reminder depends on the cognitive state of a user. As an example, if a user completely forgets about a meeting, she will not be able to participate nor contribute to a task. If a user forgets some details about a forthcoming meeting (e.g., the location of a meeting), the utility of the outcome may decrease because of tardy arrival." (Kamar and Eric Horvitz 2011)

For being more specific, Pavel et al. discuss the composition between cost and benefit for the use case of setting up a medication reminder for improving the patient's adherence in medication taking. They remark, that the simple detection that someone may have forgotten to take an evening dose of one particular drug, does not always imply that a loud alarm should ring (Pavel et al. 2010). They propose the following factors for designing a protocol for reminding:

- The reminder intensity outlining different approaches of reminding with different intensities and increasing levels of annoyance like "text / light display on medication caddy $\Longrightarrow$ text / soft beep $\Longrightarrow$ louder beep on watch $\Longrightarrow$ text message on cell phone $\Longrightarrow$ phone message $\Longrightarrow$ phone call" (Pavel et al. 2010).
- The length of time since target time, where "reminding too early or too late has a higher cost that reminding on time" (Pavel et al. 2010).
- The importance of the specific medication, where it "is more important to remind a user of a critical drug as compared with a noncritical pill" (Pavel et al. 2010).
- The context of the user where for instance "reminders that account for a user's location and availability to take the medication will be far more successful than a strictly time–based reminder" (Pavel et al. 2010).

For our more general approach, we want to use more context data than just the location and also aim on evaluating the importance of one specific event with consideration of collected context information for deciding on how and when to interrupt. As reasoning about this point of interruption appears to be an elementary point for designing context–aware reminding systems, we'll discuss about different approaches in Chapter 3.3.

When speaking of designing a reminder system in a more general way, not only reasoning is benefical. Dey and Abowd suggest the following features that such a system has to provide:

- "the use of rich context for specifying reminders, beyond simple time and location and for proactively determining when to deliver them;
- the ability for users and third parties to submit reminders;
- the ability to create reminders using a variety of input devices;
- the ability to receive reminders using a variety of devices, appropriate to the user's situation;
- the use of reminders that include both a signal that something is to be remembered and a full description of what is to be remembered; and
- allowing users to view a list of all active reminders." (Dey and Abowd 2000)

We mention these points only for the sake of completeness and for generating awareness that also different other disciplines in software and usability design need to be considered for the creation of a fully functional reminder system. For our special use case, we specialize on the usage of context data for reasoning about reminding or alerting and will therefore not discuss about approaches in how to generate, store and manage reminder data.

3.2 Context Data

We already delivered basic definitions of the term context within the area of context–aware computing at the beginning of this chapter. Here, we'll deal in more detail with the topic on how context can be categorized and acquired.

Abowd et al. draw the following relation between the design of context–aware applications and the context to decode:

> "Context–aware applications look at the who's, where's, when's and what's (that is, what the user is doing) of entities and use this information to determine why the situation is occurring. An application doesn't actually determine why a situation is occurring, but the designer of the application does. The designer uses incoming context to determine why a situation is occurring and uses this to encode some action in the application." (Abowd et al. 1999)

Based on this observation, they split context types into primary and secondary types. Primary context types involve location, identity, time and activity for answering the questions of where, who, when and what and also act as indices for other sources of contextual information, which is also stated as secondary context type:

> "For example, given a person's identity, we can acquire many pieces of related information such as phone numbers, addresses, email addresses, birthdate, list of friends, relationships to other people in the environment, etc. With an entity's location, we can determine what other objects or people are near the entity and what activity is occurring near the entity." (Abowd et al. 1999)

Within our study, we're aiming on gathering primary context data for the location, identity, time and activity using the sensors of smartphone devices. From these categories location data is rated as one of the most useful (Chang 2013). Secondary context data will also be acquired for additional information acting as further description for the connected primary context type.

Context data itself can be obtained through various ways. One approach covers the direct access of the device's sensors. In this case, the sensors are usually locally built in and the desired information can be gathered directly from the software. In a middleware infrastructure low–level sensing details are hidden and encapsulated (see Figure 10). This approach can be enhanced using a context server, where sensor data can be obtained concurrently through multiple clients (Chen, Finin, and Joshi 2003). As we're solely focusing on the development for mobile devices with built in sensors, we won't make usage of context servers and will focus on gathering data directly or by the usage of functionality provided by the Android SDK.

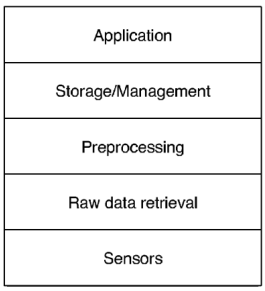


Figure 10: This figure illustrates an example of a middleware infrastructure for gathering sensor data and its associated layers (Baldauf, Dustdar, and Rosenberg 2007).

Horvitz et al. bring up examples for sensors, that can be used for gathering information about one's attention, which is amenable for us to gather information whether a reminder is missed or not:

> "Perceptual sensors include microphones listening for ambient acoustical information or utterances, cameras supporting visual analysis of a user's gaze or pose, accelerometers that detect patterns of motion of devices, and location sensing via GPS and analysis of wireless signals. However, more traditional sources of events can also offer valuable clues. These sources include a user's online calendar and considerations of the day of week and time of day. Another rich stream of evidence can be harvested by monitoring a user's interactions with software and devices. Finally, background information about the history of a user's interests and prior patterns of activities and attention can provide valuable sources of information about attention." (Horvitz et al. 2003)

Also Zulkernain, Madiraju and Ahamed arranged a survey for generalizing contexts for interruption management on mobile devices, where they come up with similar results.

They identified the user's location (divided into in–motion and static location), the user's schedule (identified by calendar info, the day of week or time of day) and the interruption feature (defined as interrupter–user relationship, interaction history with the interrupter and interruption content) as the most important ones (Zulkernain, Madiraju, and Sheikh Iqbal Ahamed 2011). These results give us some basic idea for on which context data to emphasize when collecting and evaluating our data. As the data of a wide range of sensors and contextual data listed in this section are accessible for smartphone applications, they will also be used within our study. All of the context and sensor data in usage will be discussed in greater detail in Chapter 4.2.1.

Also, Prejovic and Musolesi suggested in their research different flavors of context for interruption management with notifications on mobile devices, which are:

- "Notification context. The context in which a notification is sent determines an interruption success.
- Response context. The context in which a reaction is recorded determines an interruption success.
- Notification–response context change. In case of a successful interruption, the context changes from the notification to the reaction.
- Notification context variation. The variation of context at the notification time indicates an interruption success." (Pejovic and Mirco Musolesi 2014)

In our study, these points are the basement of our evaluation, as we're also measuring the disruptiveness of interruptions. The setup of our interruption dialog and how data will be acquired will be discussed in Chapter 4.1.

3.3 Context Reasoning

Context reasoning itself is declared as referring to "the inference process for the values of high–level, deduced context attributes from those of simple, low–level attributes" (Kumar and Xie 2012, 280). In this thesis, we concentrate on system driven reasoning with the result to decide whether it is an appropriate moment to interrupt the user (here in terms of displaying a reminder) based on gathered data about his or her context or not. Therefore, general guidelines on how to implement such a system will be discussed in this section.

When designing and developing a decision model, Horvitz and Apacible applied the following steps:

- Event and context capture, with recording of various events (e.g. calendar or desktop) and actions like the user's activity,
- tagging and assessment, where for instance the recorded data is linked to data about perceived interruptibility and finally

- generation and testing, where the tagged cases are classified and one or more reasoning algorithms are applied (Horvitz and Apacible 2003).

Godbole and Smari deliver a similar approach for deploying interruption aware systems, where data is gathered by a human perspective approach — like questionnaires and surveys — and transformed to training information for machine learning algorithms. Based on this information, interruption–aware models are created, which respond appropriately to new interruptions based on user situations (Godbole and Smari 2006). This process is outlined in Figure 11.

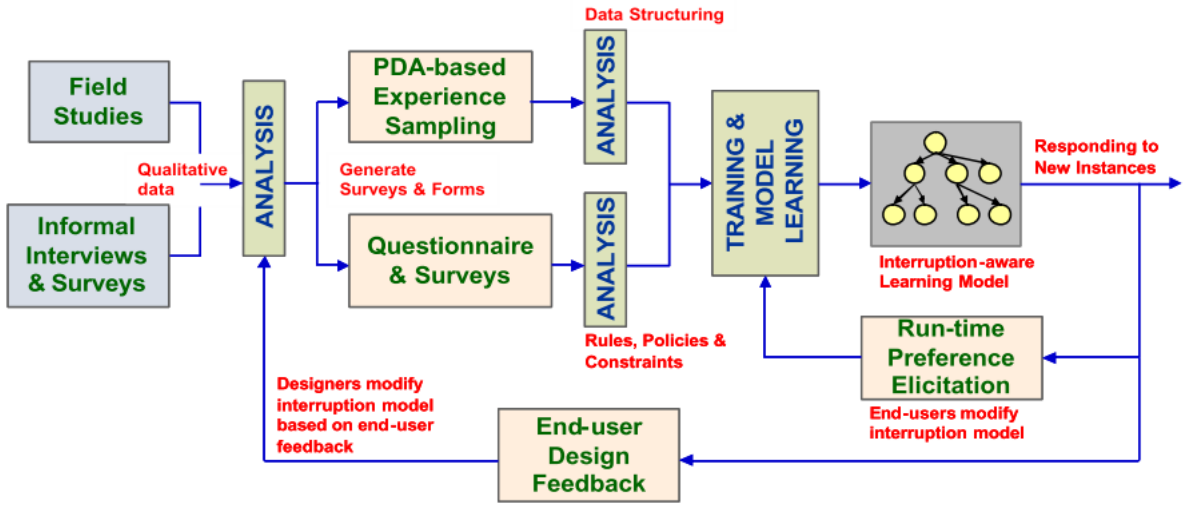


Figure 11: Here, the process of the generation of interruption–aware models is illustrated. Qualitative data gathered through studies and surveys will be transformed to structured data for training models and used as input for interruption–aware learning models (Godbole and Smari 2006).

In general, context reasoning approaches can be distinguished to rule–based, machine learning and hybrid approaches. Rule–based implementations are defined as following:

> "A rule is a declarative statement that specifies how the values of several simple or deduced attributes can be combined to derive the value of another deduced attribute. Both logic and ontology–based context models imply the use of rule–based context reasoning. The rules can express implicit semantics embodied in the model language constructs as well as explicit user–defined semantics." (Kumar and Xie 2012, 280)

Machine learning approaches focus on identifying patterns in context data and using these patterns to infer a context attribute value from a set of values of other attributes, whereas a hybrid is a combination of both machine– and rule–based implementations (Kumar and Xie 2012, 280). These strategies will be discussed in the following sections.

3.3.1 Rule Based Approaches

As already annotated in the introduction of this chapter, rule based approaches can be defined as "simple IF–THEN rules used to specify how context–aware systems should adapt" (Schilit, Adams, and Want 1994). Rule–based context reasoning therefore is defined as "an expressive way to define situations, which are crucial for the implementation of many context–aware applications" (Nicklas et al. 2008). Such situations can also be referred as preferences or context–preconditions.

Henricksen and Indulska introduce a preference model, where each preference is modelled as a pair of a scope and a scoring expression. The scope serves as a description of the context in which the preference applies. On the other hand, the scoring expression assigns a score to such a choice, which can either be a numerical value between zero and one or one of four special values, where:

- "$\natural$ represents a veto, indicating that the choice to which the score is assigned should not be selected in the context specified in the preference scope;
- $\bar{\wedge}$ represents obligation, which is essentially the opposite of veto;
- $\perp$ represents indifference or an absence of preference; and
- ? represents an undefined score, signalling an error condition." (Henricksen and Indulska 2006, 10)

An example for creating such a reasoning model with multiple preferences is illustrated in Figure 12. For calculating the score based on multiple preferences, the following approach is proposed:

> "Preferences can be grouped into sets and combined according to policies, such that a single score is produced for each choice that reflects all preferences in the set. The policies dictate the weights attached to individual preferences and determine how conflicting preferences are handled. One common policy involves averaging the numerical scores, but allowing vetoes, obligations and undefined scores to override." (Henricksen and Indulska 2006, 11)

When applying these rules to a hypothetical example for finding the perfect time for setting up a medical reminder, we can for instance set up a veto when the user is not at home, meaning that no reminders will be fired when the location is not identified as home as the veto overrides all the other scores based on different contextual data. Another example is also given within the description of Figure 12.

In order to react to context changes, Henricksen and Indulska introduce triggers, which react on state changes between true, false and possibly true. These triggers can be fired once, a predefined amount of times, between certain time intervals or always (Henricksen and Indulska 2006, 14). An example for such a trigger is represented in Figure 13.

p1= when $SynchronousMode(channel) \wedge \neg CanUseChannel(callee, channel)$
rate $\natural$
p2= when $Urgent(priority) \wedge SynchronousMode(channel)$
rate 1
p3= when $Urgent(priority) \wedge \neg SynchronousMode(channel)$
rate 0.5

Figure 12: Outlining of an example for creating a reasoning model: The first example *p1* forbids the use of synchronous channels, when there is no access to required devices; *p2* and *p3* together imply that synchronous channels are preferred for urgent calls before asynchronos ones (Henricksen and Indulska 2006, 10).

upon $EnterFalse(Occupied(\text{“Amy Carr”}))$
when *true*
do Notify of recent missed calls
always

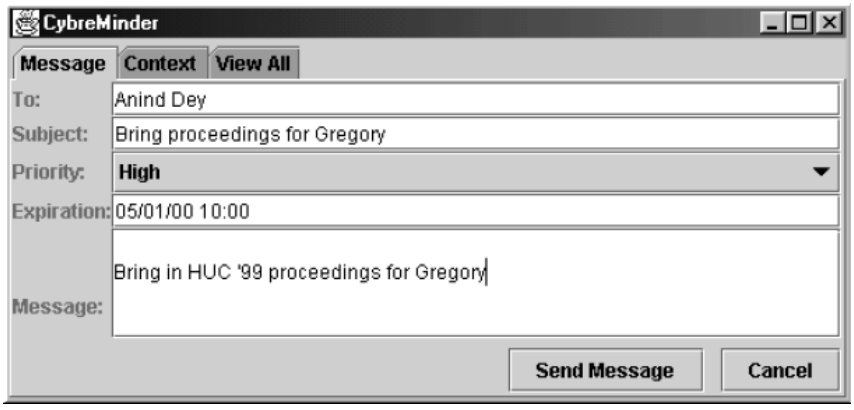
Figure 13: Outlining of an example for creating a trigger in a reasoning model: Every time the user *Amy Carr* is no longer occupied, she will be notified of recent missed calls (Henricksen and Indulska 2006, 14).

For using a simple trigger within our hypothetical example, we can for instance define a trigger on location change to re–evaluate the score values once the user arrives at home, as it removes one veto and might lead to a different outcome for the final score. So other values, like for instance the schedule depending on the calendar, might also be evaluated for deciding whether to remind or not.

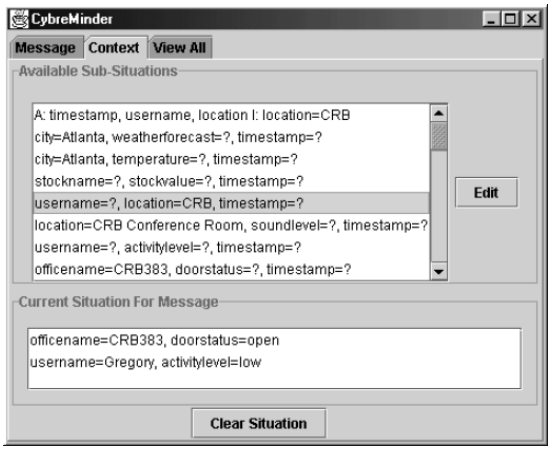
This model brings up some interesting and easy to implement ideas for reasoning within different context–preconditions. But especially for setting up context–sensitive reminders some rules appear too strict for us. So for instance a veto would always prevent a reminder to appear. When bringing up our hypothetical example again, our user would never receive a reminder if he or she would have never gone home. So instead of overwriting the other score values, the assignment of low probability values to veto conditions might turn out more useful to avoid conditions where no reminders will show up within desirable time intervals.

When laying focus on specific implementations for context–sensitive reminding tools, we bring up the example of CybreMinder, which is a tool where users can create and receive reminders based on contextual situations (Dey and Abowd 2000). Figure 14 delivers an overview over this application. Also, Schilit, Adams and Want describe a similar system setting up context–triggered actions based on predefined conditions (Schilit, Adams, and Want 1994). One big disadvantage with focus on our use case is the fact, that context data,

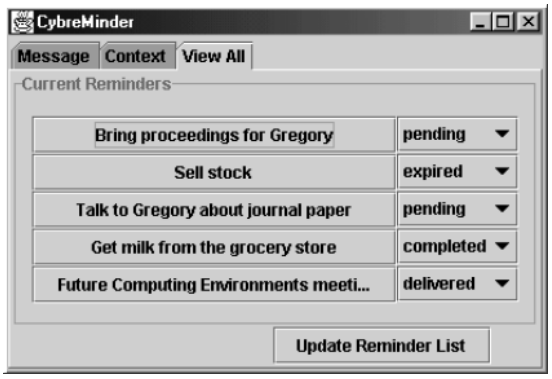
like time or locations, need to be predefined. This is a situation we're aiming to avoid for our suggestions towards implementation. Of interest for us is the status of those predefined reminders, which can be "completed" (reminder has been addressed and can be dismissed), "delivered" (reminder has been delivered but yet need to be addressed) or "pending" (reminder should be delivered again when the associated situation is next satisfied). For supporting an easy to enhance interpretation and reasoning about contextual data, they separated those mechanisms from the actual application into an own framework (Dey and Abowd 2000). As we're only collecting some basic ideas about reasoning on rule based context reasoning and not aiming on implementing a solution solely depending on this concept, we won't describe their solution any further. The main reason is the too restricted approach in predefining contextual data. Also it is mentioned, that machine learning approaches offer greater advantages within reasoning using contextual cues than static rules or case based approaches (Godbole and Smari 2006).



(a) Defining a Message



(b) Setting up Contextual Preconditions



(c) View Status of Reminders

Figure 14: First, a reminder message can be created, then contextual preconditions can be defined, and finally the state of all reminders can be supervised (Dey and Abowd 2000).

3.3.2 Decision–Theoretic Approaches

Kamar and Horvitz introduced a decision–theoretic approach for distinguishing "reminders that are beneficial for a user's performance from the ones that are not" (Kamar and Eric Horvitz 2010, 2). For achieving that, they calculate an expected value of reminding (EVR) based on the following principle:

> "A reminder for task m is beneficial for a user if the expected utility for receiving a reminder about m is higher than the cost of interruption given the current state of the user. The utility of a reminder depends on the cognitive state of a user: has the user forgotten all or some information that might be included in a reminder." (Kamar and Eric Horvitz 2010, 2)

For calculation, they come up with the following formula:

$$\begin{aligned} EVR(m) =& p(F^m|E)p(A^m|E) \\ & (U(R^m, E) - U(F^m, E)) \\ & + p(D^m|E)p(A^m|E) \\ & (U(R^m, E) - U(D^m, E)) \\ & - COI(m, E) \end{aligned} \tag{1}$$

The three states F^m, D^m and R^m are introduced and described as following:

> "(1) F^m represents the state in which a user has forgotten all about m, (2) D^m represents the state in which the user has forgotten or is unsure about a subset of details regarding the task, such as its location, start time (or deadline), and other participants, and (3) R^m represents the state in which the user remembers that task m exists and also remembers all of the details regarding the task." (Kamar and Eric Horvitz 2010, 2–3)

So, for instance $p(F^m|E)$ is the probability that the user is in the state F^m based on a given evidence E, whereas $U(R^m, E)$ presents the user's utility being in state R^m. Those two factors are connected, as like for instance when "a user forgets some details about a forthcoming meeting (e.g., the location of a meeting), her utility may decrease because of tardy arrival" (Kamar and Eric Horvitz 2010, 3). Finally, $p(A^m|E)$ represents the relevance of m to the user him– or herself by illustrating "the likelihood that the user would engage in task m" if he or she remembers it (Kamar and Eric Horvitz 2010, 3). On the other side, "*COI(m, E)* represents the cost of interrupting the user by delivering a reminder about m, given evidence E about the user's state" (Kamar and Eric Horvitz 2010, 3).

It appeared to be feasible to measure this cost as the willingness to pay in dollar values to avoid disruption in certain attention states. This approach has been used in several fields

of decision analysis and it appears that users feel comfortable with assessments in that way (Horvitz and Apacible 2003). For the calculation different ways are proposed, solely depending on the use case for the reminder or alert itself. Within the COORDINATE service, Horvitz et al. make the following suggestion for assigning this value for reminding about meetings:

> "The system computes an expectation,
>
> $$ECI = p(A^m|E,\xi)\sum_i p(c_i^m|E,\xi) + (1 - p(A^m|E,\xi))c^d \quad (2)$$
>
> where A^m is the event of attending a meeting, c_i^m is the cost of interruption associated with interruptability value i, c^d is the default cost for the time period under consideration, and E represents appointment attributes, the proximal context, time of day, and day of week." (Horvitz et al. 2002)

Also, Horvitz, Jacobs and Hovel deployed a more general formula for an expected cost of alerting (ECA):

> "Consider a set of alerting outcomes, A_i, F_j, representing the situation where a notification A_i occurs when a user is in a state of attentional focus, F_j. We assess for each alerting outcome, a cost function of the form $C^a(A_i, F_j)$, referring to the cost of being alerted via action A_i when the user is in attentional state F_j. Given uncertainty ab out a user's state of attention, the expected cost of alerting (ECA) a user with action A_i is,
>
> $$ECA = \sum_j C^a(A_i, F_j)p(F_j|E^a) \quad (3)$$
>
> where E^a refers to evidence relevant to inferring a user's attention." (Horvitz, Jacobs, and Hovel 1999)

For the development of BusyBody — a desktop program asking regularly about the user's current cost of interruption — Horvitz, Koch and Apacible come up with a quite similar formula:

$$ECI = \sum_j p(I_j|E)u(D_i, I_j) \quad (4)$$

Here, I defines the state of interruptibility and E once more the observational evidence (Horvitz and Apacible 2003).

When looking at these different ways to calculate this value, we also encounter a reoccurring pattern with the summation of the user's probability being in predefined states and the corresponding cost factors based on evidence like contextual data. As a result, the definition of a cost–function for disruptiveness of interruptions on mobile devices can most likely be handled in a similar way.

When coming back to the overall calculation of the EVR value, its final significance is stated as following:

> "It is beneficial to send a reminder to a user if the reminder is associated with a positive EVR value. If so, the expected benefit of interrupting and sending a reminder is greater than the cost of interruption associated with it." (Kamar and Eric Horvitz 2010, 3)

We already came up with similar implications in our discussion about reminder systems in Chapter 3.1. For the setup of such a system, we'll discuss the steps on implementation for one specific example and combine it with appropriate knowledge gathered from different literature.

Kamar and Horvitz used this knowledge to build up a reminding system for meetings called Jogger (see Figure 15). For setting up training data, they asked participants to label meeting entries, where they constructed a relevance model (handling the probability that a future meeting will be attended), priority model (predicting the overall priority of a meeting), memory model (the likelihoods if a future meeting will be remembered or forgotten) and interruption models (providing the cost of transmitting an alert to users). For constructing probabilistic dependency models, they generated decision trees based on contextual data as illustrated in Figure 16 (Kamar and Eric Horvitz 2010, 6). These models "are used to predict whether a meeting is important, whether a user intends to attend the meeting, whether the user has forgotten about the meeting, and the cost of interrupting the user based on her current context" and combined with the calculation for the cost of an interruption "to form the expected value of a reminder" (Kamar and Eric Horvitz 2010, 11).

For a more general point of view, Horvitz et al. describe the usage of Bayesian attentional models, where input sensors are used for gathering evidence about attention and providing the means for computing the probabilities over user's attentions and intentions. They distributed an influence diagram, representing the fusion of sensors and making of decisions in the context of an user's attention under uncertainty (Figure 17).

For deciding whether a notification is displayed or not, they introduce the Notification Platform, which serves as a layer between the incoming messages and the user (illustrated in Figure 18a). Within the design of this platform, classifiers are trained to distinguish incoming messages for their urgency (comparable to the description in the last paragraph in the use case of meetings). A schematized view of the Notification Platform is shown in Figure 18b, where the Context Server acts as the central concept of a Bayesian attention model collecting clues about attention and location. It includes a Context Whiteboard, which stores key abstractions from this evidence such as "voice trace detected", "single application focus" or "head tracked — looking away from display". This data is used not only for deciding if the user gets interrupted from his or her workflow, but also for deciding on how to interrupt (Horvitz et al. 2003). As seen in Figure 18b, their solution is designed for several devices and therefore also amenable for our study focusing on mobile applications.

For another short description on how to set up probabilistic models, Horvitz, Jacobs and Hovel describe such a setup as following:

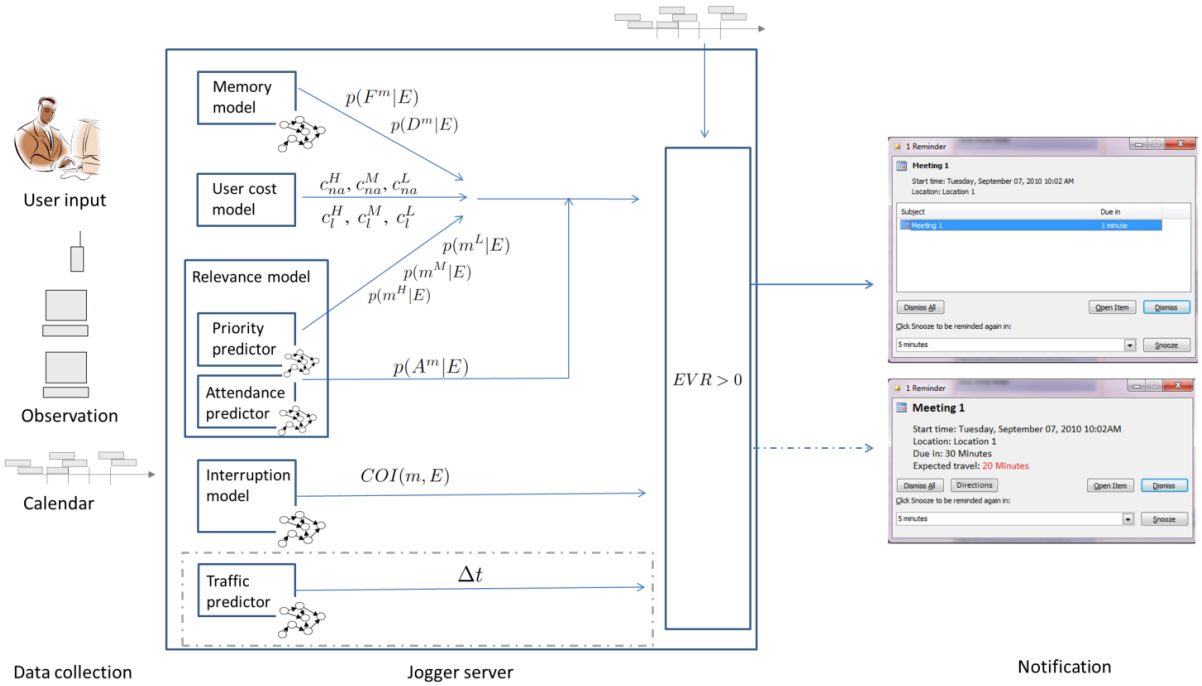


Figure 15: Here, the architecture of the reminding System Jogger is outlined. Depending on the input of various models, an expected value of reminding (EVR) is calculated. When the expected benefit of interruption is greater than the associated cost (EVR > 0), the user will be interrupted (Kamar and Eric Horvitz 2010, 4).

> "In building probabilistic models for inferring the context–sensitive cost of distraction, we consider a set of mutually exclusive and exhaustive states of attentional focus and seek to identify the cost of communicating an alert given a probability distribution over the states of a user's attention. Such states of attention can be formulated as a set of prototypical situations or more abstract representations of a set of distinct classes of cognitive challenges being addressed by a user. Alternatively, we can formulate models that make inferences about a continuous measure of attentional focus, or models that directly infer a probability distribution over the cost of interruption for different types of notifications." (Horvitz, Jacobs, and Hovel 1999)

An example for building up a Bayesian model for inferring the probability distribution over a user's attentional focus is illustrated in Figure 19. For a setup to get information on the user's availability, Zulkernain, Madiraju and Ahamed propose a system based on three tiers, which are:

- The context information tier, which aggregates information from internal sources and stores this information in a context data store, to pass it to the tree generator for processing,

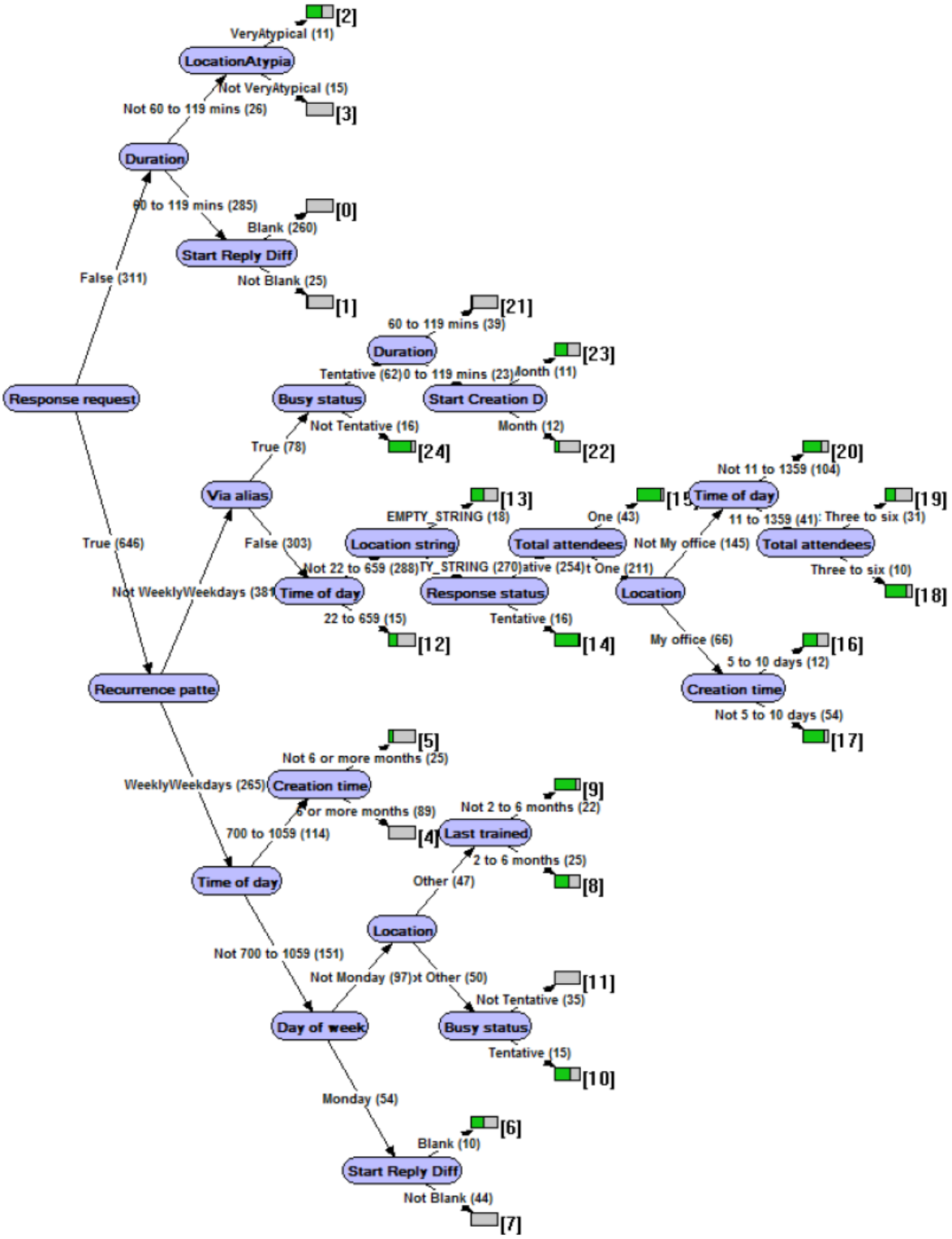


Figure 16: This decision tree illustrates a memory model "for predicting the probability of forgetting about the occurence of a meeting, based on the selected attributes of a meeting" (Kamar and Eric Horvitz 2010, 7).

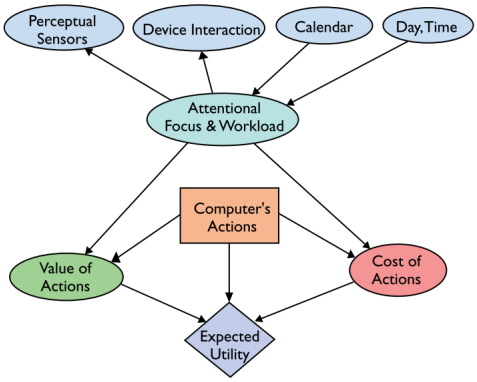
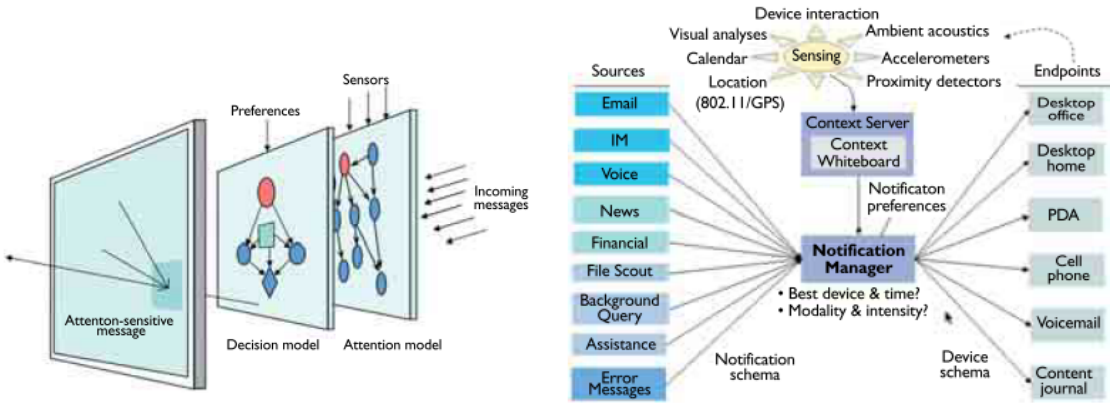


Figure 17: This influence diagram represents the fusion of sensors and the decision making under uncertainty in the context of an user's attention (Horvitz et al. 2003).



(a) Setup (b) Schematized View

Figure 18: These figures outline the setup and the architecture of the Notification Platform (Horvitz et al. 2003).

- user preferences, which includes user specific choices and rules and
- the decision tree, receiving input from both context information and user preferences for generating a decision tree (Zulkernain, Madiraju, and Sheikh Iqbal Ahamed 2010).

This architecture and an example for a decision tree used for this data are outlined in Figure 20.

To put the benefits of context–sensitive reminders into numbers, Kamar and Horvitz come up with the results, that "a context–sensitive reminder system, instead of a traditional reminder system increases the total value generated by reminder systems by 19.5 percent while decreasing the total number of interruptions by 46.6 percent" (Kamar and Eric Horvitz 2010, 15). Recent research — implementing a library for interruptions on

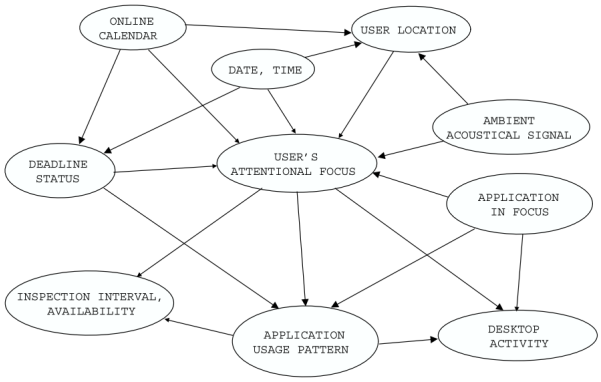
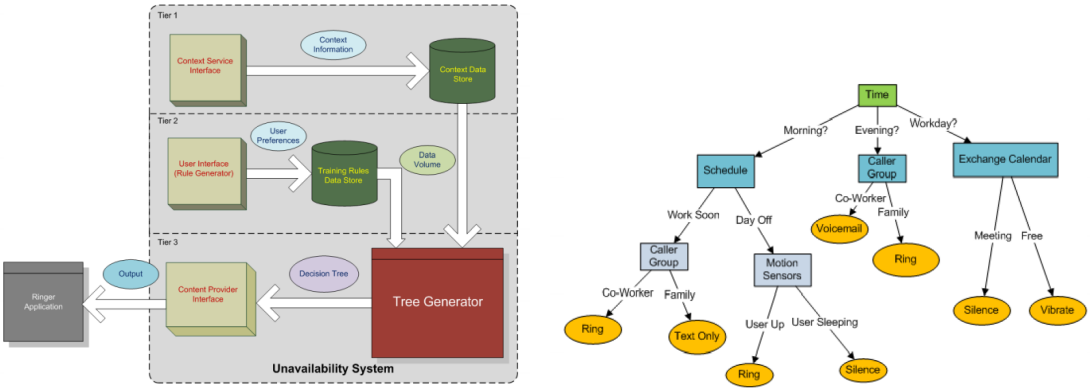


Figure 19: An example for building up a Bayesian model for inferring the probability distribution over a user's attentional focus (Horvitz, Jacobs, and Hovel 1999).



(a) Tier Layers (b) Decision Tree

Figure 20: The context information tier (aggregating information from internal sources) and user preferences (including user specific choices and rules) deliver input for generating a decision tree (Zulkernain, Madiraju, and Sheikh Iqbal Ahamed 2010).

mobile devices — also follows comparable steps by evaluating hypothetical classifiers for a machine learning approach and further individually train the algorithm depending on the user's reaction to interruptive events. Pejovic and Musolesi were implementing an interruption management library for Android smartphones called InterruptMe following these steps. They also come up with similar results as Kamar and Horvitz:

> "An extensive experiment shows that, compared to a context–unaware approach, interruptions elicited through our library result in increased user satisfaction and shorter response times." (Pejovic and Mirco Musolesi 2014)

All this research proves, that the implementation of context-aware systems leads to benefits by decreasing the amount of disruptive events in comparison to traditional approaches. Therefore, it also delivers evidence of the gain in HCI by lowering the negative effects of

inappropriate interruptions as discussed in Chapter 2. Because of that, it appears feasible for us to pursue this decision–theoretic approach in implementing context–aware reminder systems. How such systems can be practically implemented in connection with the results of our study will be further discussed in Chapter 5.

3.4 Recent Studies

Here, we will compare our approach with recent research and highlight differences and similarities to our study outlined in Chapter 4.

Poppinga, Heuten and Boll observed notifications on mobile devices to predict opportune moments based on the mobile context inferred through the phone's sensors. They conducted a long–term study over up to 76 days evaluating 6581 notifications on 79 users. Like in our study, they solely rely on sensor–based context data including timestamp, location provider, position accuracy, speed, GPS heading, compass heading, roll, pitch, proximity and light level. Around every three hours and fifteen minutes a notification is shown where the user enters details about his or her mood and the obtrusiveness of the interruption (see Figure 21). 15 seconds before the interruption occurs, context data is saved and one minute later the notification is automatically dismissed. They discovered, that time appears to be a good predictor and that the answering rate decreases significantly during nighttime. The average pitch angle was indicated as most relevant aspect in predicting whether an user reacts to a notification or not. They used the pitch angle for gathering clues if the user is holding the phone in his or her hands or not (Poppinga, Heuten, and Boll 2014).

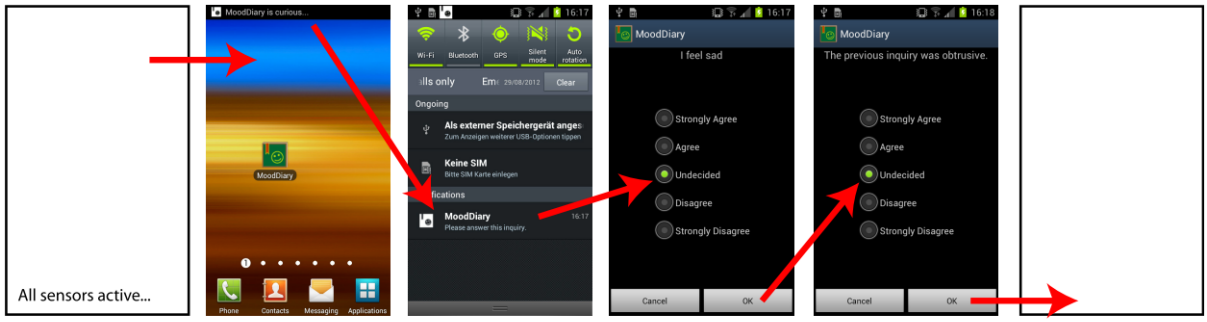


Figure 21: This figure illustrates the steps taken in the study of Poppinga, Heuten and Boll. After contextual information has been evaluated, they show a notification, where the user is asked about his or her mood and the disruptiveness of the interruption. Finally, this data is stored online (Poppinga, Heuten, and Boll 2014).

In comparison to our study, their approach is following the same basic ideas with triggering interruptions at regular time intervals and collecting contextual data. As they published their App in the Google Play store, they were able to collect data for a high number of

participants, but they also decided against collecting data of location and microphone data. In our study, the number of participants is more restricted and there is also less notification data to analyze, as our study lasted only around one week for each participant. On the other side, we're collecting more contextual data, including data that is tracked continuously in no direct connection to the upcoming interruptive event. For instance, with listening to screen on/off events, we expect to get more precise data on phone usage than with analyzing the pitch angle.

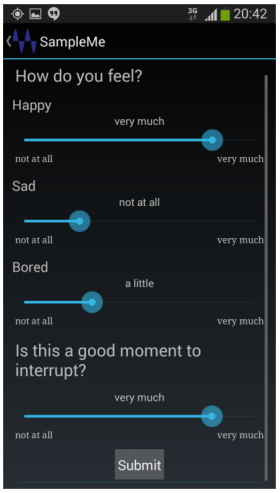
Pejovic and Musolesi also tackled the problem of designing intelligent interruption mechanisms for mobile devices. As in our study, they hypothesize, that context gathered through the phone's sensors help in determining user's interruptibility. For gathering data, they created an Android application called SampleMe, where they notify a user regularly about a survey which consists of questions about the user's attitude towards being interrupted, his location, activities, company and emotions (see Figure 22). In total, they collected 2334 notifications from which 906 were answered by 20 participants within two weeks. They also sample the user's location and label it with "residential", "work" or "public" through the survey. Based on how the user answered the question "Is this a good moment to interrupt?" (which can be "not at all", "a little", "somewhat" and "very much") they train a classifier within context changes in the reported GPS location, accelerometer features, Bluetooth and WiFi environment. They discovered, that the change in the Bluetooth environment is significantly different between opportune and non–opportune moments for interruption, whereas they couldn't find a correlation between interruptibility and activity change. Based on this work, they deployed the InterruptMe library, where they used classifiers trained by the SampleMe application to find opportune moments of interrupting. For measuring the effectiveness of their solution, they ran a study on ten participants whom received random and context–aware interruptions measured by InterruptMe through collected contextual data. Compared to context–unaware approaches they state their results as promising for future work (Pejovic and Mirco Musolesi 2014).

Within our study, we also plan on gathering clues on interruptibility based on contextual data. We are also asking a similar question for measuring the disruptiveness of the interruption and also offer the same amount of answers. We're not collecting additional information through a survey. Instead, we're gathering a wider variety of contextual data with the hypothesis that information about the user's context can be derived from this information. Also, we conducted our study on a higher number of participants with the motivation to gather more granulated data.

Pielot, Church and de Oliveira investigated the nature and effects of notifications on the daily lifes of mobile users. Within a week they collected data from 15 participants consisting of real–world notifications in combination with their subjective perceptions in an online diary. During the study, they sent an email with a link to a survey where the users were asked to reflect upon the notifications received the day before. They identified the most recent used messenger, email, social and other Apps alongside information in combination with the survey regarding subjective perceptions, attitudes and emotions and personal stories towards interruptions. They come to the conclusion, that a participant has to deal with a mean number of 65 notifications each day mostly generated by messengers.

Group	Features
Time	Time of day, weekend indicator, time into experiment.
Accelerometer	Mean, variance and mean crossing rate of the accelerometer readings. Activity variation around the notification time.[#] Activity change from notification to response.[*+]
Location	Descriptive location: "Residential", "Work", or "Public". Location change from notification to response.[*+] Bluetooth fingerprint change from notification to response.[*+] WiFi fingerprint change from notification to response.[*+]
Company[*]	Company indicator: "alone", "not alone".
Activity Engagement[*]	Descriptive activity: "Work related", "Leisure", or "Maintenance". How important is the activity? How interesting is the activity? How challenging is the activity? How skilled is the user wrt the activity? How concentrated the user is? User's desire to do something else.
Emotions[*]	How happy, sad, angry, frightened and neutral user is?

(a) Features Analyzed



(b) SampleMe Dialog

Figure 22: Here, we outline the features analyzed and the dialog used to measure this data in the study of Pejovic and Musolesi (Pejovic and Mirco Musolesi 2014).

Receiving too many emails caused negative emotions, whereas messaging or social network updates lead to positive emotions by feeling connected to others. Their results cover the conclusion, that dealing with notifications was delayed by the user when he or she was busy, but nevertheless notifications from messenger or social networks were usually viewed within minutes. As a final result, they encourage future work in finding strategies to lower perceived negative emotions towards notifications (Pielot, Church, and Oliveira 2014).

Compared to our approach, their study is much more specialized on App usage and notifications caused by them. We evaluate a broader amount of context specific data, but we also record data regarding application usage but within a different goal. Where Pielot, Church and de Oliveira focus on finding how users feel towards notifications and interruptions we try to find contextual clues on disruptiveness of events.

Böhmer et al. conducted a study for discussing design alternatives for the user interface when giving notification about incoming calls. They tackle the problem, that current–generation phones are still using abrupt full–screen notifications for alerting users about incoming phone calls, which might be undesirable when interrupting the user from an important task. For achieving this, they focused on implementing alternative interfaces for handling phone calls. In comparison to our study, the information they received about the user dealing with phone calls is of interest. They analyzed data from 525 participants with an average time period of around a month. Their results are outlined in Figure 23. They monitored that most calls were answered and out of interruptive calls more than half of them were accepted. In their implementation, they offered the possibility to postpone calls to continue using the active App, which has been used to finish micro–interactions before allowing the interruption (Böhmer et al. 2014). This observation covers one of our discussions in Chapter 2.2.3 where we identify the predictability of interruptions within task switching as an efficient strategy for lowering their disruptiveness.

	#cases	#users	per user
Incoming calls total	**88,516**	**525**	**168**
... non-interruptive	**59,608**	**519**	**114**
... interruptive	**28,908**	**525**	**54**
Interruptive calls accepted	**16,119**	**509**	**31**
...after being postponed	**106**	**79**	**1**
Interruptive calls declined	**2,311**	**317**	**7**
...after being postponed	**114**	**78**	**1**
Interr. calls unanswered	**10,476**	**468**	**149**
...after being postponed	**539**	**206**	**2**
Postpone events	**770**	**247**	**3**
Widget move events	**3,048**	**403**	**7**

Figure 23: Here, we offer an overview of the study results of Böhmer et al., tackling the problem interruption management on incoming phone calls (Böhmer et al. 2014).

Due to their different focus, Böhmer et al. were not elaborating different contextual data. Nevertheless, they deliver us insight into phone usage by analyzing data about phone calls. As we're also recording data about past phone calls for finding clues about the disruptive nature of interruptions, their observations justify our decision for taking this data in account.

Pielot also dealt with the topic on how users react to phone calls but in comparison to Böhmer et al. he followed a context–aware approach. He conducted a study to investigate to what extent sensor and contextual information on mobile devices can predict phone call availability. For gathering data, he published an App called Silencer in the Google Play store, where users can mute incoming calls by shaking the phone. Along 418 users and 31311 calls 15 features have been logged, which are:

- "the ringer mode and when it last changed,
- charging state and when the phone was last (un)plugged,
- the screen state (on/off) and when it last changed,
- the day of the week,
- the hour of the day,
- the proximity sensor (display covered/not covered),
- the pitch angle of the display,
- the level of acceleration right before the call,
- how often the same caller had called before,
- the time since the last call,
- whether the last call was picked up,
- whether the last call was silenced,
- and whether the user took this call (ground truth)." (Pielot 2014)

Within classifiers for generic or personalized models, he reached a high accuracy in predicting the user's availability. Pielot claimed, that the user activity and an approximation of daily routines are the strongest predictors for deciding whether a user will pick up a call or not (Pielot 2014).

Compared to our approach, Pielot's study is aiming specifically on an individual handling with phone calls. As a result, the contextual data recorded is also brought into specific connection to this use case. Our study is less specific by trying to find opportune moments for interruption in general. As we're also recording charging and screen states for gathering knowledge about user activity, we'll evaluate if we come to comparable results in the importance of these factors.

When looking at these recent approaches in collecting and analyzing data about user behavior on mobile devices, we come to the conclusion, that a lot of research in this area is currently in progress. We identified a few contextual groups that have been measured as important throughout the studies discussed in this section. Especially location data, information about phone usage and the time when interruptions occur have been identified as strong classifiers for the disruptiveness of notifications. When analyzing the results of our study in Chapter 4.3 we will also look out if we come to similar results and also if we are able to determine additional patterns on recorded contextual data of the users participating in our research.

3.5 Synopsis

In this chapter, we discussed about context–aware computing and reminding within mobile applications. We delivered definitions for the terms context and context–aware systems and outlined their basic requirements and feasibility within the area of mobile devices. The advantages of context–sensitive reminder systems and features for improving their performance were evaluated.

Context data was analyzed in more detail and categorized into primary and secondary types, whereas location data was rated as one of the most useful (Chang 2013). How context data can be obtained has been discussed briefly in addition to the types of context useful for mobile devices.

Within context reasoning, different approaches were discussed and their utility has been compared towards our implementation suggestions. Rule based approaches were evaluated and the conclusion was drawn that veto values which overwrite multiple data of evidence are not feasible for generating context–aware reminder systems. Also, the need to predefine contextual rules was defined as avoidable. Machine learning approaches were identified as delivering greater advantages and discussed in greater detail within the section of decision–theoretic approaches.

In the section of decision–theoretic approaches, the calculation of an expected cost of reminding was discussed with the basic principle, that the benefit of interrupting must exceed the cost of interruption. Different ways of calculating the cost of interruption were

listed and reoccurring patterns, like the summation of the user's probability of being in a specific state with the corresponding cost factor based on evidence, were identified. In a specific example, the steps for implementing reminding systems were listed and the usage of probabilistic models was discussed, identifying Bayesian models and decision trees as a useful solution. As the benefits of decision–theoretic approaches appear beneficial to our use case, we'll focus on giving suggestions for an implementation of a reminding system using this knowledge later on.

In the end, we looked into recent studies following comparable approaches and highlighted similarities and differences to our study. We come to the result, that phone usage, location and daily routines were identified as strong predictors for the disruptiveness of interruptions on mobile devices. Therefore, we used this comparison as motivation for the next chapter for finding out if we can come up with comparable insights. Here, we'll aim on conducting data to identify important contextual patterns in connection to interruptions and their disruptiveness on mobile devices.

4 Study on Contextual Data

To identify appropriate moments for interruptions and reminding, we developed an application for the Android platform, which causes alerts at regular time intervals. The main motivation is to collect contextual data using the phone's sensors and estimate their significance in comparison to the disruptiveness of the interruptions created by the application. Our application has been installed on various Android phones of users participating to this study. Data was gathered by using the phone's internet connection to store it to an online database. Within this chapter, we'll describe and discuss the setup and also the outcome of our study. First of all, we describe and evaluate the course of actions the participant has to go through from the beginning to the end of the study. Next, we'll discuss the overall software architecture from developer side with special focus on contextual data, answering which data is collected why and how. After the setup of our study has been outlined, we deliver an overview over the data collected and interpret it according to the alerts and their evaluated disruptiveness. This interpretation serves as the foundation for our final discussion and implementation suggestions within the last chapter of this thesis.

4.1 Course of Actions

We set up a website[8] delivering basic information about the study, the data which is gathered, contact details for further questions and both a QR–Code and a link leading to the APK file, which needs to be installed on the Android device. Based on the language of the operating system running on the device, the application is either displayed in German or English language. Here, we mostly refer to the English terms that occur throughout the usage.

After installation, the user needs to enter an email address for offering us a possibility to get in touch with him or her and a few demographic details, from which we think that there might be an impact on how an individual perceives interruptions. These data include sex, age and occupation. As we aimed on keeping the participant's hassle for starting the study low, we decided to offer a choice of predefined entries.

For gathering the participant's age, we offer entries for

- 16 to 24 years,
- 25 to 34 years,
- 35 to 44 years,
- 45 to 54 years,
- 55 to 64 years and

8. `http://cmayr.smarthealth.at/` (accessed on November 11, 2014)

- 65 years and older.

If age has an impact on how interruptions are interpreted, we believe that these categorizations inherit enough significance as we discussed earlier that the human perception is more depending on the tasks performed (see Chapter 2). Nevertheless, we are collecting this data to possibly find different usage patterns within different age groups (for the case the entered data is diverse enough).

To categorize the participant's occupation, we divided between

- fully employed,
- part–time employed,
- self–employed,
- in education,
- unemployed and
- prefer not to say

We think, that the occupation delivers a difference in the participant's daily routine and might influence one's awareness towards interruptions. For instance, we bring up the hypothesis, that a fully employed individual has a more regulated daily routine (in terms of working hours) than a student with a non–periodic schedule.

Finally, a predefined rest time can be adjusted. Initially, we set this time from 10 pm to 10 am, meaning that no interruptions will be triggered within this period. We didn't keep track of users changing their resting times specifically, as we gather such information indirectly by the distribution of times when alerts are triggered throughout a day. After setting up all this data, the participant can start the study, which minimizes our App and sets up an icon on the notification tray indicating that a background service is running.

From now on the user will be interrupted around every two hours (including a random variance of plus or minus fifteen minutes). This interval is influenced by the studies of Czerwinski, Horvitz and Wilhite, where they conducted a diary study of knowledge workers characterizing how people interleave multiple tasks within interruptions. In their results, they reported 0.7 interruptions in a task lasting for average 53 minutes (Czerwinski, Horvitz, and Wilhite 2004). When recalculating these values for one interruption, we come up with a task duration of approximately 75 minutes. As a result, we come to the conclusion, that an hourly interruption for our reoccurring alarm might be evaluated as too annoying. As three hours appeared to us as a too long time interval for keeping track of the participant's daily routine, we decided to set our interval to approximately two hours.

When the user is confronted with an interruption caused by our implementation, he or she will see a dialog, which is illustrated in Figure 24. For on how this dialog makes itself

noticeable, we orient on the ringer mode set by the user for being consistent to his or her current preferences on how being interrupted in general. We distinguish between the following cases:

- Silent mode, where no alarm sound and no vibration occurs,
- vibrate mode, where only vibration occurs and
- ringer mode, where the standard alarm sound will be played.

Within the appearing dialog, the user should measure the disruptiveness of the occurring interruption. One of the most crucial things for gathering usable data is the process in finding the right question to ask in combination to the answers to evaluate. Therefore, we analyzed current literature for avoiding ambiguous interpretation.

Borg and Staufenbiel identify rules for the formulation of questions and answers (defined as items) supporting the retrieval of clear and statistically processable results. Therefore, an item should be

- clear and compact,
- understandable for anybody by avoiding unneeded complex phrases,
- free of double negatives,
- addressing only one specific topic,
- formulated in a neutral way and
- able to be interpreted in an unambiguous way (Borg and Staufenbiel 2007, 16).

Finding the right question is one of the most important tasks for our study, as the chosen answers are keystones for our final evaluation and results. We decided to formulate a question which follows all the rules defined by Borg and Staufenbiel with emphasis on easy, fast and unambiguous comprehension. Finally, we came up with the following question in German:

Wie empfinden Sie die Unterbrechung?

For Android phones using a different system language, we translated the same question to English:

How do you perceive the interruption?

It can be seen, that we avoid any formulation of possible negative effects of the message causing the interruption (like for instance by asking "How disruptive is this interruption?") to guarantee a neutral impact to the user when deciding upon an appropriate answer.

For the answer, we decided to choose Likert–Items, where the user can choose between four predefined options ranging from one (for not disruptive) to four (for very disruptive). We

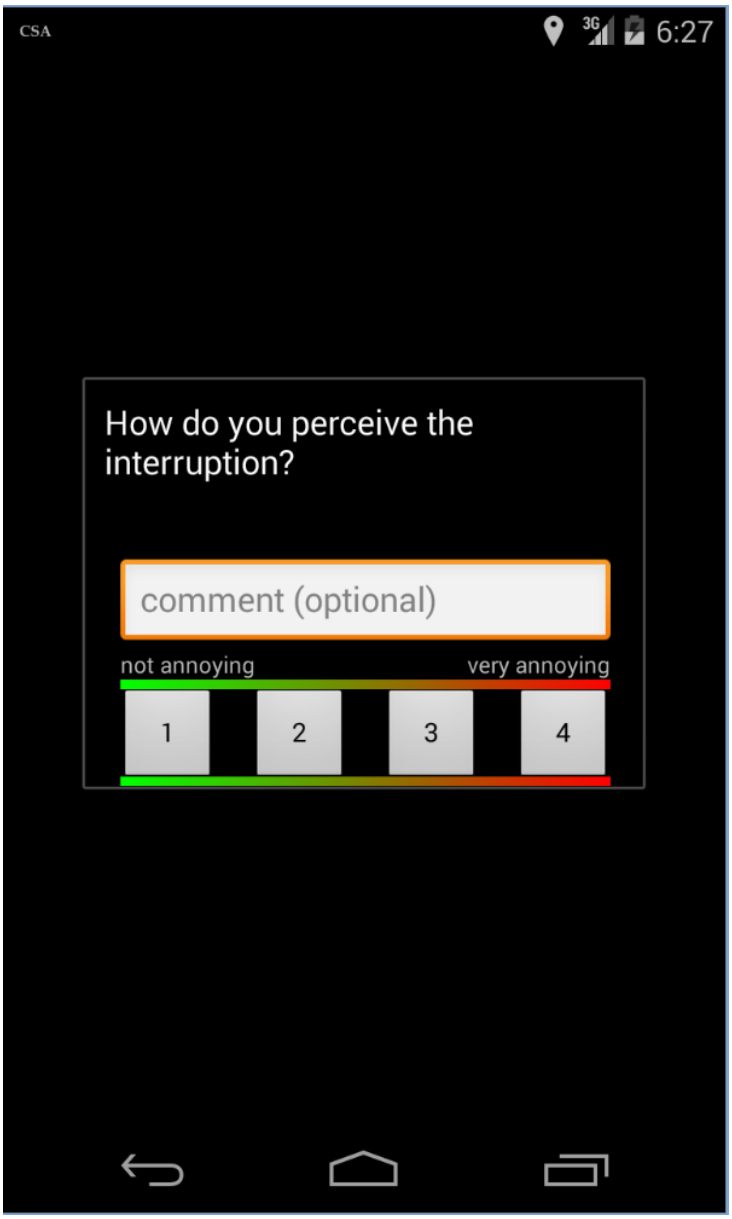


Figure 24: Here, we show the appearance of the alert dialog on the user's phone. The user should rate the disruptiveness of the current interruption from a scale from one to four and can also add an optional comment. Additionally, the tray icon CSA (for Context–Sensitive Alerts) — indicating that the background service of our App is running — is visible.

decided to use this format because such a model is usually easy to comprehend (Borg and Staufenbiel 2007, 21). For our use case, this leads to the advantage, that the participant has to spend less time dealing with our artificial interruption. Also, exactly four options are used because of the following reasons:

- We want to deliver a more accurate distinction between the disruptiveness of an interruption. Only two options (disruptive/not disruptive) would be too coarse for our evaluation and three options lead to the possibility that one might feel too restricted upon his or her choices (Borg and Staufenbiel 2007, 26).
- Studies prove that people can store around seven (plus or minus two) distinctions in their working memory (Miller 1956). According to that, more choices than that would be simply overwhelming for human beings.
- The usage of an even amount of choices allows no neutral answers and forces the user to decide either for approval or rejection (Borg and Staufenbiel 2007, 22). That's a behavior we want to achieve, as we only want to have information about a "neutral" interruption when there is no reaction within a certain time frame. Also, we want to categorize the interruption itself between disruptive or non–disruptive as we're aiming to bring context data in connection to these answers where a neutral reaction would be of no great help.
- Also in terms of the limited display space we're sticking to four possible answers instead of for instance six. In terms of usability and readability within different possible phone setups this amount of button appears to us most promising. Other studies also suggest that it is rarely necessary to use more than five different levels of numerical preference ratings (Henricksen and Indulska 2006).

Beside the four possible results, we also evaluate a fifth outcome, which is rated as "not recognized". When there is no response to the alert for a certain time interval, it will close itself and send the phone back to standby mode. We decided to use an idle time of two minutes within the participant has the possibility to respond to the alert. This choice is inspired by the study of Pielot, Church and de Oliveira where they logged notifications and the subjective perception upon a group of participants on mobile devices. They collected data hinting that messenger notifications are usually viewed within 3.5 minutes on weekends and even less on weekdays (Pielot, Church, and Oliveira 2014). These results are also outlined in Figure 25. We draw connections to messenger and social network notifications, as they also tend to be immediately viewed due to their limited content.

Additionally, we offer the possibility to enter an individual comment in the alert dialog for giving the participant the opportunity to deliver feedback about special situations within an interruption may occur (like for instance circumstances perceived as really annoying). We don't expect to receive much data through this option though. The main motivation behind this idea is to offer a more advanced channel of communication to the user, which can but doesn't have to be used. All in all, we focused on designing the alert dialog in a way, that the user can comprehend and respond to the interruption in a fast way for minimizing possible errors that can be caused by such artificial breaks (McFarlane and Latorella 2002, 15).

While the user is interrupted that way in regular time intervals, we collect different contextual data, which is either directly connected to the alert or continuously stored. This data will be outlined in greater detail in Section 4.2.1.

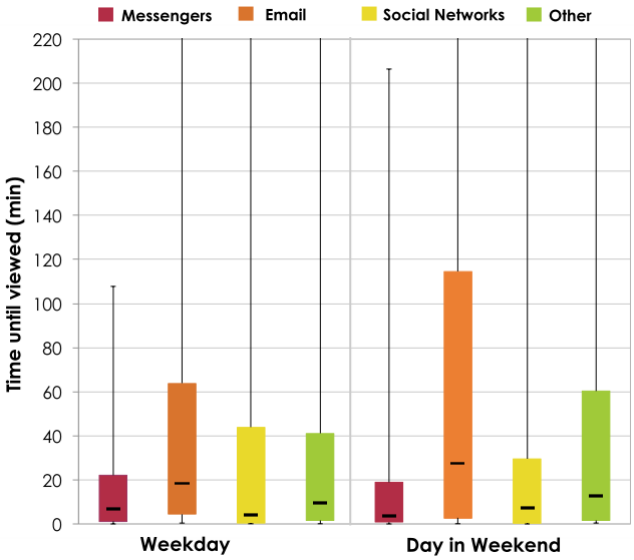


Figure 25: The study of Pielot, Church and de Oliveira gives hints that messenger and social network notifications are usually viewed within 3.5 minutes on weekends and less on weekdays. Nevertheless, this group of notifications has the lowest response time in average (Pielot, Church, and Oliveira 2014).

In order to keep the different data in terms of amounts of interruption comparable between different participants, we decided to limit the length of the study to the number of interruptions and not to a predefined time interval. As a result, we decided to collect data for 30 individual alerts as we are aiming on mirroring the mobile usage of around a week. When taking the default values for the predefined resting time in account, around five to six alerts are triggered per day, which means that the study is finished after around five to six days. Of course, with shorter periods of resting time, more alerts are triggered per day and therefore the study would be finished sooner (and vice versa with longer periods of resting time).

When 30 alerts have been fired, the study is finished and a thank you message is prompted to the participant, also displaying whether all data has been sent to our servers or not (see Figure 26). When all the data has been successfully transmitted, the App can be uninstalled.

4.2 Architecture

After discussing the course of actions on user side, we'll outline the technical background and which data is stored. The App itself has been developed natively in Java for Android devices. We were aiming on providing support for a wide variety of devices to make this study accessible to a potentially high number of participants. Current statistics show that within October 2014 more than 10 percent of devices are still running on Gingerbread

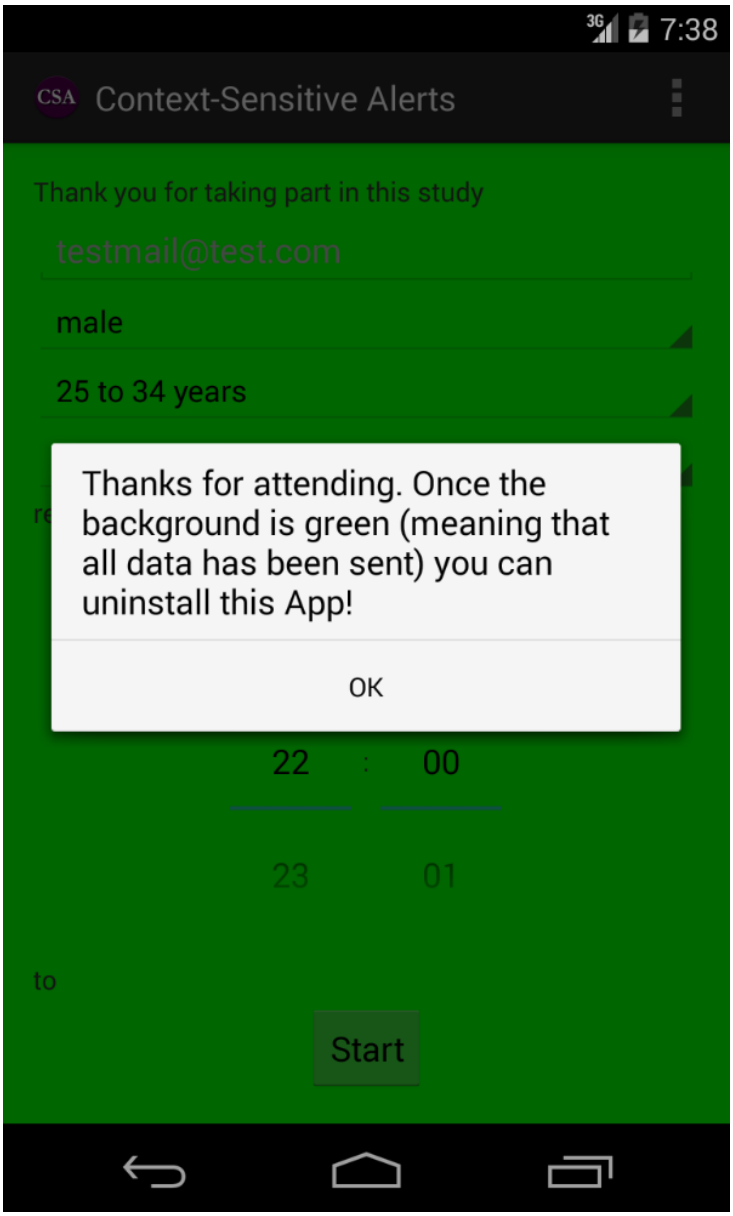


Figure 26: The end of study will be communicated to the participant by a thank you message. When the background is green, he or she can uninstall the App.

(versions 2.3.3 – 2.3.7) and bellow[9]. As a result, we made our implementation working on platforms down to Froyo (version 2.2) to follow our aim of high compatibility. Within this section, we will first discuss which data is collected (with emphasis on contextual data) and then how this information is processed.

9. http://www.statista.com/statistics/271774/share-of-android-platforms-on-mobile-devices-with-android-os/ (accessed on November 12, 2014)

4.2.1 Collected Data

All data is stored online on a MySQL database. When an online saving fails (due to no internet connection or various other reasons) the information will be saved on the phone's internal storage. On change of the phone's connectivity status (which also happens on restarting the device) an attempt in sending yet unsent data will be started. This overall behavior can be reconstructed in the activity diagram in Figure 27.

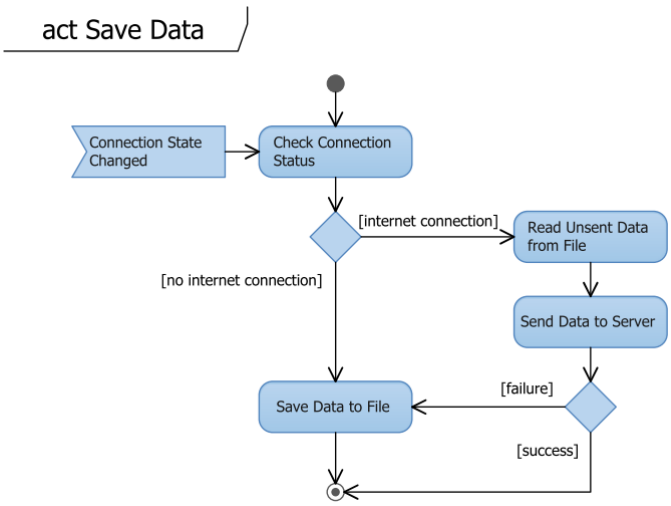


Figure 27: When there is no internet connection, unsent data will be stored in a text file. Otherwise, all unsent data will be sent to an online server and removed from the queue on success.

Except for data about the alert or the user, all collected information is contextual based on data provided by the phone throughout its usage. Beside age, occupation, gender and email (as already discussed in the previous section) we also store the following information for each participant:

- The country code of the SIM card for gathering coarse information about the user's nationality and also cultural background,
- API level and device name for overall statistical information on the phone,
- the join date for information about the study length and
- the phone's IMEI number for identifying the participant and allocating the contextual data stored.

Beside result, ringer mode, comment and data (also discussed in the previous section) every triggered alert includes the following additional data:

- The battery level and whether the phone is charging or not for indirect information on the phone's "availability" (low battery level might lead to standby, charging state may be an indicator that the phone is currently locally bound) and

- once more an unique ongoing ID for identification in combination with the participant's IMEI number.

When talking about the contextual data evaluated within our study in greater detail, we will separate it into two categories. First of all data, which is stored and evaluated once an alert occurs, and then data, which is collected periodically throughout the length of the study.

Data in Connection to an Alert Studies regarding context–aware computing on mobile devices hint, that the location is evaluated as one of the most useful context (Chang 2013). As a result, the evaluation of location data appears essential to us. Due to privacy issues towards the participants of our study, we are aiming on avoiding the collection of personalized data. Therefore, we don't store specific location points. For being more specific, we start tracking the location of the user around fifteen minutes before an alert occurs and only store the maximum distance covered between all recorded location points till the alert is being displayed. With this value we aim on gathering information about the "movement rate" of the user right before an interruption. A high value offers a clue for fast transportation, like for instance the usage of a car or train, where a medium value might deliver a hint to physical activity like walking or running.

In past studies, audio in context to interruption management has for instance been used to get clues about a person's availability (Horvitz et al. 2002). Also it has been used to track more specific events like conversations (Horvitz, Koch, and Apacible 2004). A few minutes before an alert occurs, we start tracking the maximum amplitude measured by the phone's microphone in regular intervals. Right before we trigger an interruption, we calculate the average amplitude based on the recorded data and store this value. As result, we aim on collecting evidence on the sound level of the user's environment.

The last data, that we record during an alert, are entries in the user's calendar. As an example, Horvitz and Apacible used information of the user's online calendar for deciding whether a person is in a meeting or not (Horvitz and Apacible 2003). We extract information of the Google calendar and store for each alert all the events that are listed during that time including start and end time. This approach is highly restricted, as we only get information about possible appointments when they are listed in the user's online calendar. Nevertheless, we rely on the Google calendar as source for information, because we believe that this is a standardized way within the Android infrastructure and its inherited Google services, where we can access this information throughout all Android versions programmatically.

Continuously Tracked Data While conducting our study, a background service is launched on the participant's phone with the purpose to collect data in the background for gathering information about the phone's usage. While accelerometer data is recorded periodically, other information is recorded depending on certain triggers. An overview is

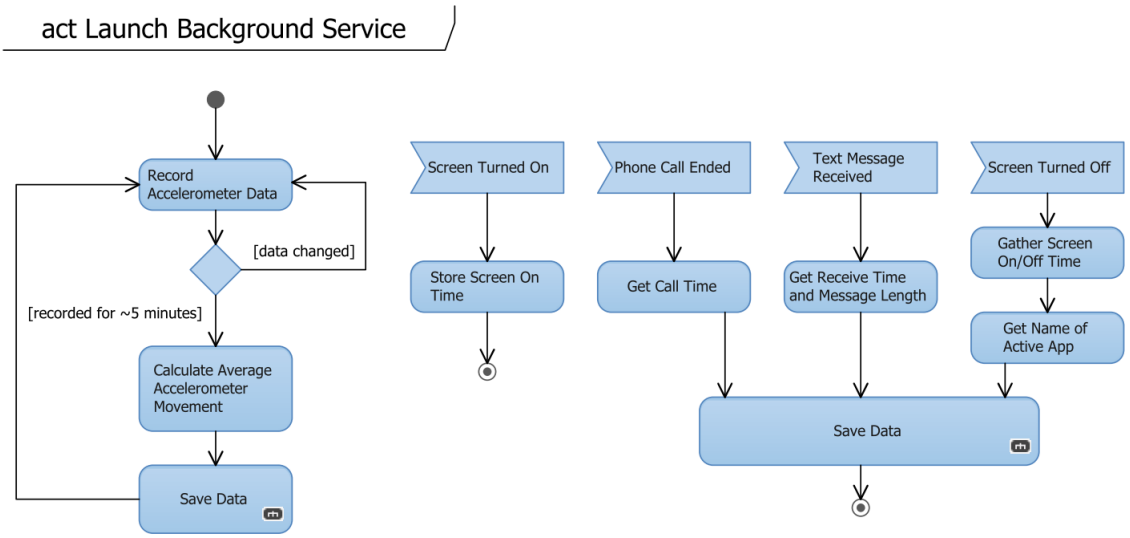


Figure 28: The average data of accelerometer movement is periodically stored; screen on and off events give us information about phone usage patterns; data of phone calls and received text messages are also stored directly.

illustrated in Figure 28. Which type of data is recorded at which cause will be discussed in the following paragraphs.

Previous studies already proved, that the evaluation of accelerometer data can identify the user's activities with a high rate of accuracy (Kwapisz, Weiss, and Moore 2011). We follow a more basic approach by using the accelerometer data for identifying whether the user is in motion or not (assuming that the phone is in his or her pocket for gathering movement data). For keeping the amount of data stored in our database low, we decided to continuously track the accelerometer's x–, y– and z–axis, but only store the average resultant distance within a time period of around five minutes. For achieving this, we calculate the square roots of the sum of the values for each axis on axis change. This calculation is also demonstrated in Listing 1. We tested our approach under various circumstances and came to promising results in movement detection. One specific example is outlined in Figure 29.

For gathering data about phone usage, we store the time once the screen has been turned on. When the screen is turned off again (triggered by the user or the device by going back to standby), we calculate the time the screen was turned on and store it with the name of the App running at this time. By doing so, we aim on finding proof for the frequency of phone usage and a possible connection to disruptiveness of interruptions. Ferreira et al. highlighted usage statistics on mobile devices with the result, that around 40 percent of application launches last less than 15 seconds. Also such micro–usage is influenced by location and purpose and most likely happens when the user is alone (Ferreira et al. 2014). Other resources report comparable results, hinting that "50 percent of mobile phone engagements last less than 30 seconds, and 90 percent of engagements last less than 4 minutes" (Yan et al. 2012). When taking this knowledge into account, we expect

Listing 1: UpdateService.java: The function getDistanceTo returns the vector distance between the current and the previous accelerometer data

```
31 class AccelerometerData {
32     float[] data;
33     float time;
34
35     public float[] getData() {
36         return data;
37     }
38
39     public float getTime() {
40         return time;
41     }
42
43     public double getDistanceTo(float[] otherData) {
44         return Math.sqrt(
45             (data[0] - otherData[0]) * (data[0] - otherData[0]) +
46             (data[1] - otherData[1]) * (data[1] - otherData[1]) +
47             (data[2] - otherData[2]) * (data[2] - otherData[2]));
48     }
49
50     public AccelerometerData(float[] data, float time) {
51         this.data = data;
52         this.time = time;
53     }
54 }
```

short periods of phone usage, hinting that the participant is nearby his or her phone and that such occurring micro–usage might also be an indicator for the user being alone.

Beside this information we also store the time of the last phone call and also the time of the last text message received including the length of the received message. It has to be mentioned that this approach is also limited, as we're unable to survey incoming messages from other (online) providers like for instance WhatsApp or Facebook. The length of the text message is recorded to receive a basic idea on how much time the user will spend reading this message(s). For phone calls, we're both listening to in– and outgoing calls for collecting additional hints about phone usage. Here, we also avoid the collection of personal data; no details about the persons, whom the user is communicating with, and no specific details of any conversations are stored.

4.2.2 Information Processing

As we highlighted the data, which is gathered, in the last section, we will now discuss on how this information is being processed. On application launch, the background service

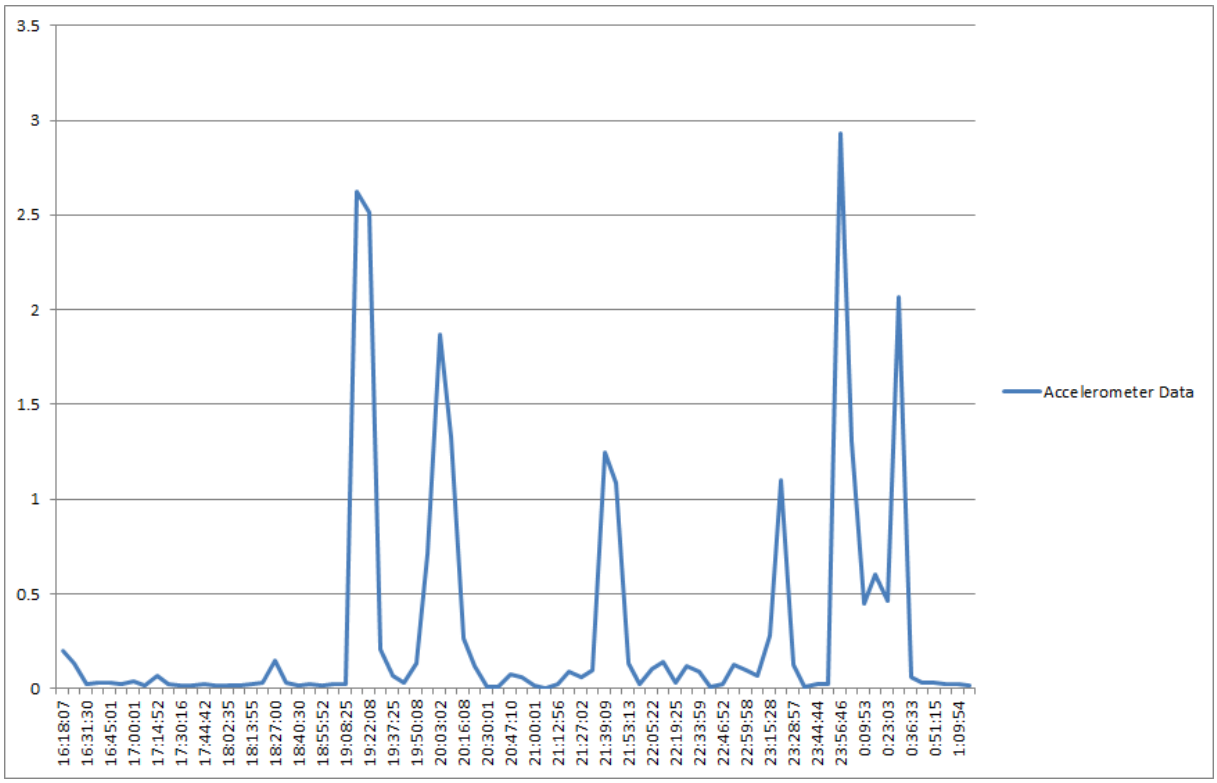


Figure 29: Here, we measured the accelerometer distance values for evaluating their expressiveness in practice: This example covers a visit to the cinema. Out of the maxima of this plot it can be measured, that between 19:10 and 19:20 we were on our way to the cinema building, around 20:15 on our way to the cinema hall, where the movie lasted from around 20:15 to 21:40 (the low movement data indicates that we were sitting); finally, the maxima around midnight indicates that we were on our way home at this time.

for continuous data collection (which has already been discussed in greater detail in the previous section) is initialized. Also the user inputs his or her demographic details (the course of actions discussed in Section 4.1) before the core functionality by setting up and triggering alerts is initiated (see activity diagram in Figure 30). When rebooting the phone, both the background service and the initialization of alerts are relaunched (illustrated in the activity diagram in Figure 31).

The core functionality of our study takes place when setting up and triggering alerts. It is outlined in great detail in the activity diagram in Figure 32 and will be discussed within this paragraph. Before initializing various alerts, the amount of already triggered alerts is checked and if it exceeds 30 alerts, the study is finished. Otherwise, three alarms are initialized in parallel, which are:

- An alarm for launching the alert, which is displayed to the user (will be triggered around two hours later including a random offset),
- an alarm for initializing location tracking (will be triggered around 15 minutes before

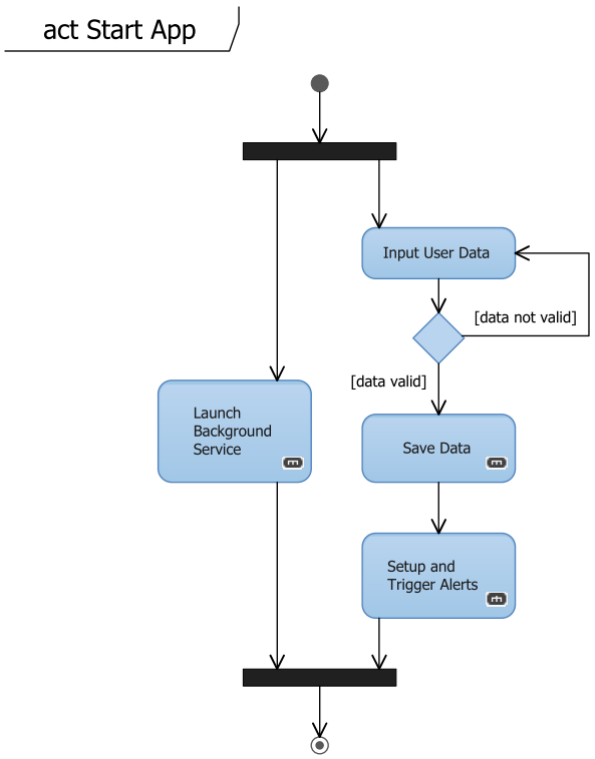


Figure 30: On application launch, the background service is initialized. After the user enters his or her data, the appropriate alerts are set up.

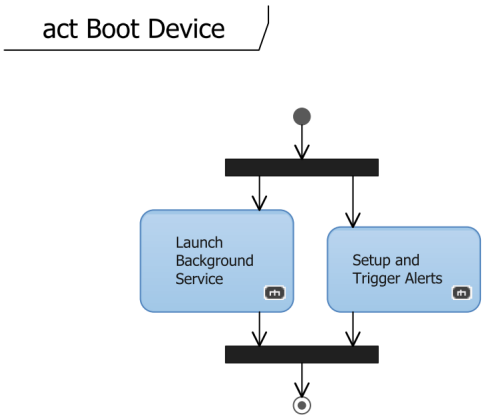


Figure 31: When the device is booting (or restarting), the background service and the alerts are re–initialized.

interrupting the user) and

- an alarm for initializing audio tracking (will be triggered around five minutes before interrupting the user).

How audio and location tracking work in detail has already been discussed in the previous

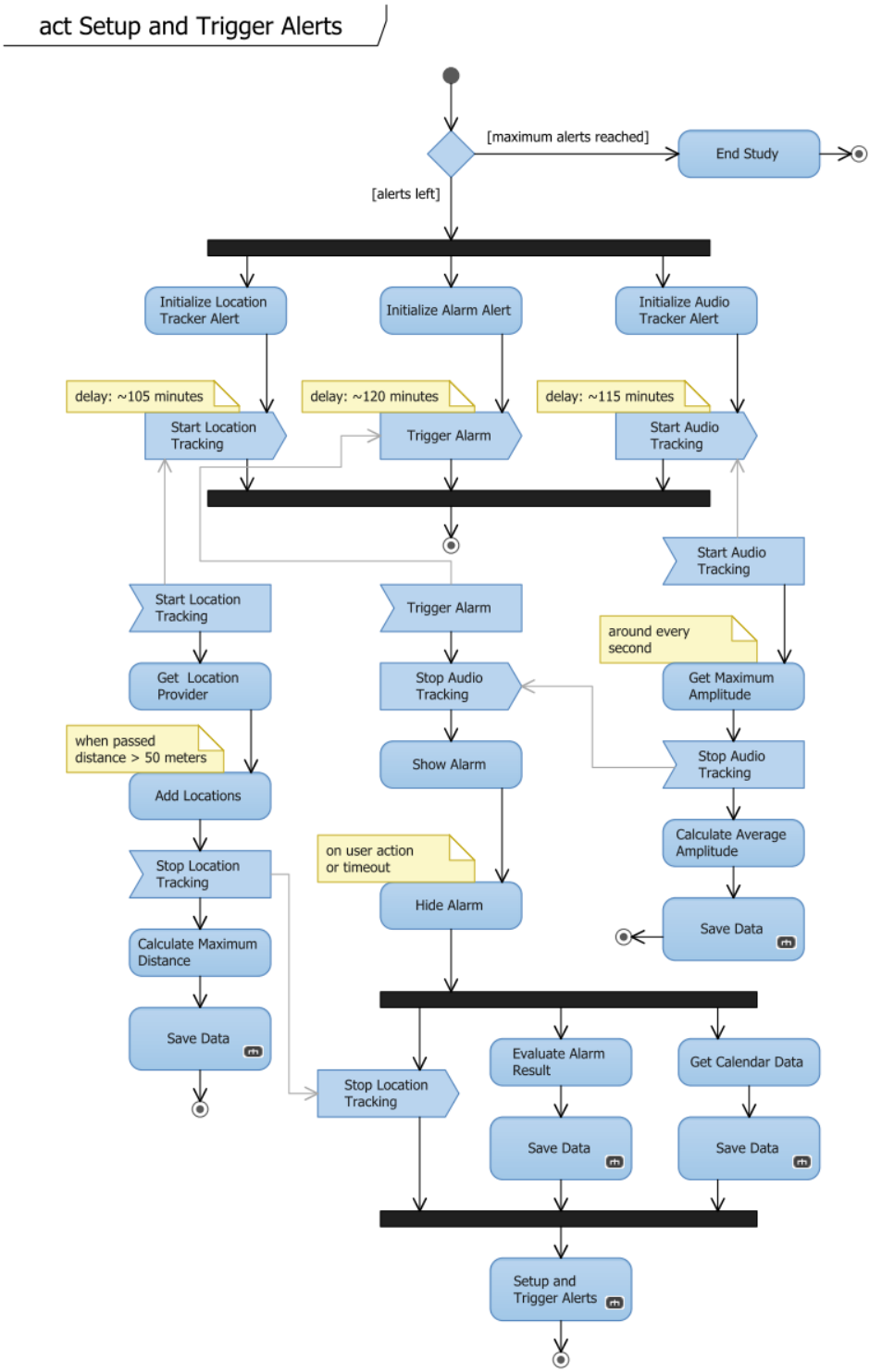


Figure 32: As long as the study is not finished, three alerts are initialized: location and audio trackers are firing before the actual alarm is triggered. When the user is interrupted, data for location, audio, alarm and calendar entries are evaluated and stored.

section. We only mention the detail fact, that audio tracking is being stopped before showing the alert for not distorting data (by for instance recording the alert sound itself or recording surrounding noises when getting the phone out of the pocket because of the interruption). After the alarm has been displayed to the user, alarm, location and calendar data are stored and the whole procedure is restarted (re–initializing these three alerts or ending the study depending on the amount of alerts already triggered).

4.3 Results

We conducted data from 32 distinct users (24 male, 8 female). 20 persons specified their age between 25 to 34 years, followed by ten persons between 16 and 24 years and two persons from 35 to 44 years. When taking this demographic data in account, the biggest user group appeared to be students (17 participants stated "being in education") followed by fully employed (9 participants), part–time employed (5 participants) and one self–employed participant. This demographic data is also outlined in Table 1. When taking the country code of the SIM card in account, almost all participants appeared to come from Austria. Only five persons had phone numbers from different countries, being the Netherlands, South Korea, Great Britain, Denmark and Bulgaria.

Age	**Amount**	**Occupation**	**Amount**
16 to 24 years	10	fully employed	9
25 to 34 years	20	part–time employed	5
35 to 44 years	2	self–employed	1
45 to 54 years	0	in education	17
55 to 64 years	0	unemployed	0
65 years and older	0	prefer not to say	0
Sum	32	**Sum**	32

Table 1: Here, we outline the demographic details regarding age and occupation which have been stated by our 32 participants.

Throughout the study we received data for 844 alerts with 554 amplitude values, 433 distance data and 62 calendar entries. We didn't receive location data for every alert, as it would've been required to turn location services on from user side for granting access to GPS or network provider data. Also, we couldn't gather audio data for each alert. We assume, that some phone models don't allow proper audio recording or access to microphone data when being in standby mode. From continuously tracked data we acquired 23829 accelerometer movements, 7613 phone activity events, 809 phone calls and 452 received text messages.

We checked up the time when a user joined the study and the last time we received an entry for accelerometer data to calculate the time on how long someone attended the study and rounded the results by minutes. Therefore, the range from the attended time has been identified from one day and 18 hours (giving us a hint that the application got

most likely uninstalled before finishing the study) up to ten days and 14 hours. In average, study data has been conducted for the interval of five days and 7 hours.

When checking the alarm data in general, we come up with the fact, that in only 20 alerts the possibility to add a comment has been used. From these comments only two contained useful data; one declaring that the participant is in a meeting and one criticizing the alarm sound, which has been mistaken for an incoming text message multiple times. As providing a comment was a non–obligatory task, we already expected not receiving much data through this channel.

Of much more interest are the alarm results and their correlations when gathering hints about the phone's availability. From all alerts, around 40 percent were declared as missed, around 20 percent as not annoying and around 14 percent as very annoying. The rest was quite equally measured as in–between. When the phone is charging, we expected an increase of missed alerts, which has been confirmed in our results. Out of 113 alerts, which have been triggered when the phone was charging, more than 50 percent were missed, almost equally followed by not annoying (18 percent) and very annoying (17 percent) interruptions. As expected, the number of missed alarms increases when only taking the alerts in account, that have been triggered in silent mode, where the interruption hasn't been indicated by a vibration or an alert sound. In this case, from 116 alerts almost 70 percent were declared unnoticed with the rest being evaluated tendentially more annoying. All these results are also outlined in Table 2. As a first indication, it appears to be favorable to trigger interruptions, when the phone is not in silent mode. When it is charging, the chances, that notifications or reminders are missed, are also slightly higher than in normal usage.

	missed	**1**	**2**	**3**	**4**
Overall	345	169	102	107	121
Charging	58	20	8	8	19
Silent	79	13	3	5	16

Table 2: When looking at the alert results (1 indicating not annoying and 4 indicating very annoying), the percentage of missed alerts increases, when the phone is in charging or silent mode.

For analyzing and portraying the relationships between further collected contextual data, we decided to plot our results in a 2D graph. For doing so, we programmed a small C# WPF application, accessing our collected data through a MySQL database connection and using the OxyPlot library[10] as supporting tool for illustrating our findings.

We already mentioned, that Poppinga, Heuten and Boll indicated in their studies, that time is a good predictor for the answer rate and disruptiveness of interruptions (Poppinga, Heuten, and Boll 2014). Inspired by their results, we decided to plot the overall amount of alert results throughout a day by counting their occurrences within one hour (see

10. `http://oxyplot.org/` (accessed on November 12, 2014)

Figure 33). Due to the predefined resting time where no alerts occur, there's of course not much response data over night, but we can draw the conclusion, that the least alarms missed are triggered in the morning before 9 am and also in the evening after 9 pm. Interesting is the observation, that we discover a peak of perceived annoyance at 2 pm, whereas 3 pm appears to be a good time for interruption as the amount of not annoying alerts is dominating at this time. Another peak for non–disruptive interruptions can also be found around noon. The later in the afternoon, the more annoying appears an alert as the amount of not annoying alarms is declining. In the evening, we register another peak for non–disruptiveness. Specifically 10 pm occurs as a perfect time for reminding or interrupting, as the amount of not annoying alerts is the highest of all results (including missed alerts). As a final result, we identify interruptions triggered in the morning, around noon, around 3 pm and in the evening as opportune moments. Especially reminders set up in the mornings or evenings appear to have a high chance of response. Therefore, we assume that the morning, noon, the middle of the afternoon and the evening are good moments of interruption as the user might take breaks from his or her regular tasks and is therefore more receptive towards artificial interruptions.

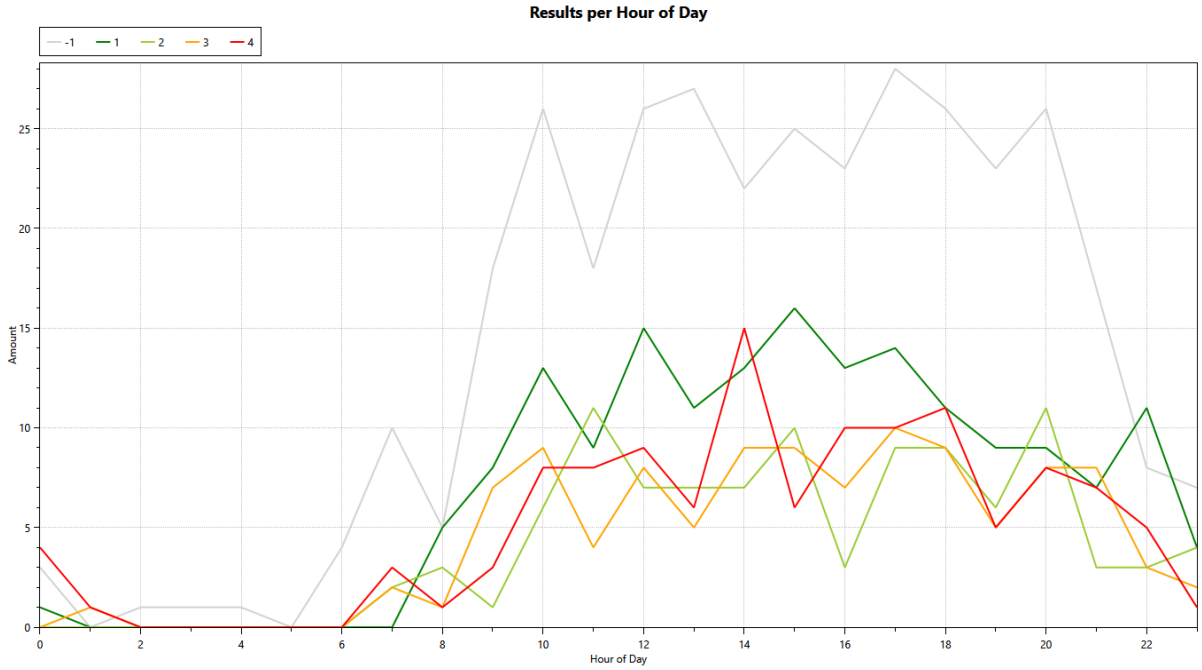


Figure 33: We plotted the occurence of alerts results for each hour of the day (where the result -1 is equivalent to missed, 1 indicating not annoying up to 4 indicating very annoying). Of interest is the peak around 3 pm, where much more alerts were rated as not annoying than very annoying. Also morning, noon and the evening can be identified as opportune moments for interruption.

When concentrating on general phone usage statistics, we already discussed studies where the short and fragmented nature of phone usage time was highlighted (Yan et al. 2012; Ferreira et al. 2014). As a result, we counted how often the user turned his or her phone on

before an alert occurs and grouped these numbers by the alert's results (starting with the evaluation of the last alert). To filter out the general occurrences within alert statistics (e.g. most alerts where missed, so the overall count of phone events in connection to missed alerts is expected to be the highest), we also summed up the time the phone was active and averaged it by usage times. As the time between alerts can vary due to resting times or also events, where the phone is turned off, we filtered these numbers again by only taking phone usage statistics in account happening one hour before an alert occurs. When looking on the unfiltered data represented in Table 3, we come to the conclusion, that interruptions were evaluated as not or slightly not annoying (result 1 or 2) after a high average phone usage compared to durations from quite and very annoying (result 3 or 4). This might be a promising result at first sight (indicating that frequent phone usage is an indicator for non–disrupting interruptions), but when taking our results with focus on alert time in account, indicating that mornings and evenings are opportune moments for interruption, we draw the conclusion, that these higher numbers are also a result of a longer evaluation period due to resting hours. When looking at the filtered data, only taking phone activity one hour before an interruption in account, we see that low phone usage is an indicator for very annoying, missed or not annoying interruptions. All in all, here the differences in average phone usage don't seem diverse enough for finding clear evidence for the disruptiveness of a future interruption.

	missed	**1**	**2**	**3**	**4**
Usage Times Between Alerts	2825	1466	825	936	1059
Active Time Between Alerts	85:50:58	43:24:27	26:37:33	18:50:10	26:18:03
Average	1:49	1:47	1:56	1:12	1:29
Usage Times One Hour Before	689	569	307	291	384
Active Time One Hour Before	15:09:50	12:34:15	7:20:32	6:56:42	8:10:16
Average	1:19	1:20	1:26	1:26	1:17

Table 3: When looking at average phone usage duration one hour before an alert, we can't find any clear patterns for the disruptiveness of an upcoming interruption, as the data doesn't appear to be diverse enough.

When taking the results of Böhmer et al. regarding phone calls in account, it appears to us that their observations regarding response time are a hint for frequent phone usage (Böhmer et al. 2014). As in general usage statistics, we counted the amount of phone calls occurring right before an alert and grouped these numbers again by the alert's results. For calculating the average value, we divided the amount of phone calls through the overall amount of alerts occurring with the corresponding result. We applied the same procedure to the amount of incoming text messages. As seen in Table 4, a low average amount of phone calls appears to be an indicator for missed or not annoying interruptions. Also a low number of incoming text messages appears to us as hint for missed alerts. We don't see a clear pattern for predicting disruptive interruptions, but generally speaking, a low rate of phone calls and incoming text messages increases the possibility to miss notifications or reminder messages within a hypothetical reminder system on time.

	missed	1	2	3	4
Phone Calls One Hour Before	105	47	55	50	47
Average	0.3	0.28	0.53	0.46	0.39
Text Messages One Hour Before	36	29	13	22	22
Average	0.08	0.17	0.13	0.21	0.18

Table 4: Less phone calls appear to be an indicator for missed or not annoying interruptions. Also, a low number of incoming text messages appears to be a hint for missed alerts.

We also applied the same strategy by summing up the average accelerometer distances for each alarm result. When looking at the averaged results in Table 5, we come to the conclusion, that the phone is slightly less moved when an alert is missed or declared as not annoying. But overall we cannot identify a clear tendency, as the values appear too similar to us. With taking the variance in measurement between different phone types in account, it appeared to us more feasible to check out this value for each participant individually. Location and audio data has also been identified as contextual data, which we are aiming to check out individually, as we're looking for certain peaks which won't be clearly identified by averaging values. As the amount of recorded calendar data is also limited to 62 entries, we will also check them out by user.

	missed	1	2	3	4
Sum of Average Accelerometer Distance One Hour Before	141.7	66	50.4	49.5	49
Amount	295	143	90	82	95
Average	0.48	0.46	0.56	0.60	0.52

Table 5: When looking at the average movement data before an alert, we can't find a clear connection to the alert result. When accelerometer data changed, we calculated the vector distance to the x–, y– and z–values of the last recorded accelerometer data. This accelerometer distance has been summed up and averaged by the amount of recorded accelerometer data.

For analyzing recorded data individually by user, we plotted all data gathered for each user in a 2D graph. On the x–axis we express the study length, starting with the join date and ending with the last date entry gathered through the accelerometer data. The y–axis is aligned to alert results, starting with zero (for missed alerts) and ending with one, two, three and four for representing the individual alert results. The values for amplitude, covered distance and accelerometer distance will be normalized for this scale. More specifically, the maximum value for amplitude and accelerometer distance will be normalized by user to the maximum scale entry of four, whereas the covered distance will be normalized by the overall maximum value through all users. We made this distinction, because we believe that amplitude and accelerometer values can have different outcomes depending on the phone's sensors. For distance data, we only wanted to visualize if the

user moved a lot or not. As we wanted to avoid misinterpreting minimum movement, we oriented to the overall movement value. For better visualization, we included all evaluated 2D plots in the appendix of this thesis.

In Chapter 3.2 we already identified location as a primary context type with high priority. Also, in our prior discussions about studies dealing with contextual data on mobile devices, location has been identified as one of the most useful contextual data (Chang 2013). When analyzing the plots of individual users, we observed relevant location changes before alerts between 13 individual users. All in all, we discovered 40 peaks within our continuous tracking of location changes. The interesting observation is, that within all of these peaks only two alert results occurred: Either the alert was missed or it was rated as very annoying. We outlined this data in greater detail in Table 6. As a result, location change appears to be a significant predictor for the disruptiveness of an interruption or the effectiveness of a reminder.

User	**Missed**	**Rated Very Annoying**	**Sum**
1	1	1	2
2	2	0	2
3	0	1	1
4	6	1	7
5	3	0	3
6	5	0	5
7	1	1	2
8	2	0	2
9	8	0	8
10	1	0	1
11	3	0	3
12	2	0	2
13	1	1	4
Sum	35	5	40

Table 6: When looking at peaks in location change, only very annoying and missed alerts occur.

For getting a feeling for the amount of location change, we ordered the passed distance values within their corresponding alert values by size. The highest value was registered as around 14 kilometers. We made the observation, that almost all alerts with location changes of one kilometer and more were missed. When only taking location changes starting with 100 meters in account, most values were registered between 200 and 800 meters. Based on these observations we can draw the assumption, that alerts or reminders shouldn't be fired right after a recorded location change of around more than 200 meters within 15 minutes.

We already outlined, that calendar data can deliver information about the user's daily schedule (Horvitz and Apacible 2003; Zulkernain, Madiraju, and Sheikh Iqbal Ahamed

2011). When analyzing our plot, we ignored calendar entries, which lasted a day or longer. We could extract 14 calendar entries lasting from one to two hours from six individual users. In Table 7 we deliver an overview over the measured values. It has to be mentioned, that in some circumstances more than one alert result could be connected to a calendar entry. We ordered these results in chronological order, only taking those in account, which occurred 30 minutes before, 30 minutes after or during the scheduled calendar event. As it can be seen within the values in Table 7, a lot of alerts triggered during such calendar entries where either missed or rated with high annoyance. Only within two exceptions we got "not annoying" (once) and "slightly not annoying" (twice) as result. Nevertheless, both of these values were recorded right after an appointment, which is giving us clues that the user might have finished his or her schedule earlier and therefore being receptive to interruptions again. As a result, we draw the conclusion, that alerts or reminders triggered during an online calendar event lasting for one to two hours are either missed or perceived as very annoying. Therefore, it should be avoided to interrupt the user during such circumstances.

User	**Number of Entries**	**Length (in minutes)**	**Measured Results**
1	5	60	4, -1
		60	3, 2
		60	4, 3
		90	-1, 4
		120	4
2	1	60	-1
3	1	120	-1, -1
4	1	60	-1
5	5	120	-1
		90	-1
		90	-1, 4
		60	-1
		60	-1
6	1	120	4, 1

Table 7: Here, we outline short online calendar entries (lasting from one to two hours) for each user and the measured results during such an appointment. This data leads us to the conclusion, that alerts or reminders triggered during such an online calendar event are either missed or perceived as very annoying.

Values, which are left for evaluation and also visualized in our 2D plot, are the periods of phone activity, the time of taken phone calls and incoming text messages, accelerometer data and the recorded amplitude values. We already analyzed user behavior based on data of phone activity, phone calls and text messages by querying data of our database. Nevertheless, we were unable to extract any reoccurring patterns by visual analysis of the plotted results. Within all of these data the connected alert results didn't appear to be in any connection. For instance, we expected to receive more annoying or missed alerts

during times of high user movement (recorded through accelerometer data) or a high amount of missed alerts within noisy environments (recorded through high amplitude values). None of these assumptions appeared to be true. We recorded regular user movement throughout the whole day without registering any peaks at certain time amounts. This uniform data couldn't be drawn in any connection to the interruptions caused by our application. For amplitude values, we assume that for instance noisy environments couldn't be recognized correctly all the time, as it might be damped when the user is holding the phone in his or her pocket. For finding clues regarding amplitude values we suggest additional studies with comparable conditions of sound recording like for instance by the usage of external microphones or by forbidding putting the participant's phones into pockets during recording.

4.4 Synopsis

In this chapter, we outlined the corner points and results of our study. First of all, we described the course of actions taken from the user's perspective. We justified our choices taken throughout the implementation of the application, like for instance the interval of the alerts triggered or the decisions taken for the design of the alert dialog. Special focus was taken on discussing the choice of the question asked when interrupting and the amount of answers provided. Also decisions regarding study length were brought up and debated.

Next, we looked into the study setup from a technical view by bringing up which contextual data is stored and evaluated. We created activity diagrams for visualizing different behaviors, like for instance how contextual data is saved or continuously tracked within a background service. We categorized the data stored into data, which can be brought into direct connection with an alert and data, which is continuously stored, with the motivation to indicate overall knowledge over the user's phone usage.

Finally, we discussed the results gathered within our study and drew conclusions, which will serve as foundation for our final discussion in this thesis. First of all, we come up with obvious observations, such as that more notifications are missed when the phone is in silent mode or charging. We identified the morning, noon, the middle of the afternoon and the evening as good moments of interruptions. We used 2D plots for analyzing the user's behavior individually. As a result, we also came up with strong evidence of location changes and short calendar entries for evaluating the disruptiveness of upcoming interruptions. Yet we were unable to find a clear connection between alert results and the data used to abstract user behavior like phone usage or accelerometer data.

5 Discussion

In the past chapters we discussed the advantages of and reasons for context–aware approaches. Throughout our study we identified contextual patterns for judging about the disruptiveness of interruptions. In combination with our discussion about implementation approaches for context reasoning we laid the foundations for answering the research questions defined in the introduction of this thesis.

As discussed in Chapter 3.2, Horvitz et al. identified valuable contextual clues by gathering data of the user's online calendar, considering the day of week and time of day or by monitoring the user's interaction with software and devices to find patterns of activity and attention (Horvitz et al. 2003). Additionally, Zulkernain, Madiraju and Ahamed emphasized the user's location, schedule and interruption feature as most important contextual data (Zulkernain, Madiraju, and Sheikh Iqbal Ahamed 2011). When taking the findings within comparable studies in Chapter 3.4 in account, we come up with comparable results. Poppinga, Heuten and Boll identified the average pitch angle of the phone (delivering indirect information about phone usage) as most important factor in predicting whether a user reacts to notifications or not (Poppinga, Heuten, and Boll 2014). Also Pielot declared user activity and an approximation of daily routines as strong predictors for deciding whether an user picks up a call or not and also gives us indirect information on how someone reacts to interruptions (Pielot 2014). On the other hand, Pejovic and Musolesi weren't able to find a connection between activity change and interruptibility (Pejovic and Mirco Musolesi 2014).

Within our study, we were able to find indirect clues about phone usage patterns and reachability. When creating a reminder system, the moment when the phone is charging has been identified as not opportune, as more than 50 percent of alerts were missed in that state. Also, when the phone is in silent mode, much more notifications (almost 70 percent) were missed than usual. Due to previous findings, we took a look into the time of day, where times around 3 pm, noon, early morning and late in the evening (right outside the resting time specified by the user) turned out as best moments of interruptions. We hypothesize, that within these moments the user is taking breaks from his or her daily routines. When taking data of phone usage in account, we only come up with clues indicating that a low rate of phone calls and incoming text messages increase the possibility of missed notifications or reminders. Plot analysis identified location change as a strong predictor for the disruptiveness of interruptions. When a location change of more than 200 meters within 15 minutes before an alert has been recorded, the alert was either missed or rated as very annoying. As a result, these circumstances should be avoided when firing interruptions. Also, calendar entries lasting from one to two hours led to a slight increase of interruptions missed or rated as very annoying.

When drawing a final conclusion to the theoretical findings inherited from other sources, it has to be mentioned, that we were able to justify the significance of location and calendar data. Also information about the current time of day has been validated as useful. On the other hand, we were unable to find clear clues between user's activity patterns and interruptibility (as in the studies of Pejovic and Musolesi).

In Chapter 3.3 we already delivered strategies in implementing context–aware systems, which we separated into rule based and decision–theoretic approaches. Here, we'll propose a suggestion on how the results of our study can be used to implement a context–aware application for reminding. Of course also different approaches tackling this problem are valid; we concentrate on finding a solution for an implementation by taking the findings of the last chapters into account.

We would pursuit both decision–theoretic and rule based strategies. Based on the clear evidence found for the disruptiveness of interruptions on location change and short calendar entries, where alerts were either missed or rated as very annoying, we can employ a rule to delay a reminder, when movement has been tracked within the last few minutes (within our test results it would be more than 200 meters 15 minutes before the interruption occurs) or when a short calendar entry can be found in the user's online calendar (within our test results this length has been defined from one to two hours). Also, the importance or criticality of the reminder regarding the user's needs should be taken in account for deciding whether and how long to delay. This value could for instance be defined by the user with values like "earliest time to remind" and "latest time to remind". The longer the time period between these two timestamps, the more space for delaying actions would be available. It can also be implied, that the urgency of a reminder increases, the nearer it moves towards its expiration. In order to guarantee that the reminder will be triggered in the user defined time period, it might be an expedient strategy to lower the exclusive strictness of these two predefined rules with increasing urgency.

When the contextual rules for location change and the user's schedule give us evidence, that the moment for interruption and reminding might be appropriate, we can alter the patterns we discovered within the decision–theoretic approaches of Horvitz in Chapter 3.3.2 based on our observations. It has been stated, that for triggering an interruption, the utility of a reminder should be higher than its cost, which can be illustrated as the summation of probabilities of the user being in predefined states and their corresponding cost factors. We were unable to gather clear evidence of the user's daily activities out of phone usage statistics. As a result, we couldn't derive direct evidence about the user's state using the phone's sensors. Nevertheless, we could find patterns giving us indirect hints about the user's availability: When the phone is charging, in silent mode or a low amount of phone calls and incoming messages has been observed, the probability to miss incoming alerts (and also reminders) is much higher than usual. Therefore, we can set up user states based on these events, which are increasing the associated cost value. As all of these single circumstances have a different impact on the amount of missed messages, they should be evaluated differently with appropriate weight values. Based on our study results, silent and charging state lead to the highest amount of missed interruptions. On the other hand, evidence for opportune moments of interrupting can be derived from the time of day, where we identified 3 pm, noon, early morning and late in the evening as moments where the user is receptive towards interruptions. Based on this evidence we could lower the cost value for reminders triggering around these time intervals. Of course, this approach is highly general, based on evidence discovered through patterns found within the results of our restricted amount of participants. We already outlined within analysis

of the Keystroke–Level Model in Chapter 2.1.3 that complex interactions are handled differently by distinctive user groups. Adopting such knowledge of individual usage to our approach means that for instance the weight values predefined through the results gathered from our study could be adapted individually based on the user's response to reminders. As a result, classifiers could be further trained based on the reaction to a reminder (e.g. a "remind me later" button or missed reminders for evaluating not opportune moments).

For future research, our results could be used as starting point (along with results covering comparable approaches listed in Chapter 3.4) for a practical implementation of a reminder or alerting service on mobile devices. Past research covering context–aware implementations led to promising results towards usability and unwanted interruptions in comparison to traditional approaches (Ho and Intille 2005; Kamar and Eric Horvitz 2010; Pejovic and Mirco Musolesi 2014; Pielot 2014). Therefore, a question to answer for such a future implementation might not be if but how much a user will benefit in terms of productivity and usability. Nevertheless, we've shown that this problem is — especially with focus on mobile devices including a vast amount of different sensors and online data — a topic, which is definitely of interest for future research and could aid in improving user's everyday life towards minimizing unwanted disruptions.

List of Figures

Listings

List of Tables

References

Abowd, Gregory D., Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. 1999. "Towards a Better Understanding of Context and Context–Awareness". In *Handheld and ubiquitous computing,* 304–307. Springer.

Adamczyk, Piotr D., and Brian P. Bailey. 2004. "If Not Now, When?: The Effects of Interruption at Different Moments Within Task Execution". In *Proceedings of the SIGCHI conference on Human factors in computing systems,* 271–278. ACM.

Adamczyk, Piotr D., Shamsi T. Iqbal, and Brian P. Bailey. 2005. "A Method, System, and Tools for Intelligent Interruption Management". In *Proceedings of the 4th international workshop on Task models and diagrams,* 123–126. ACM.

Altmann, Erik M., and J. Gregory Trafton. 2004. *Task Interruption: Resumption Lag and the Role of Cues.* Technical report. DTIC Document.

Baldauf, Matthias, Schahram Dustdar, and Florian Rosenberg. 2007. "A survey on context-aware systems". *International Journal of Ad Hoc and Ubiquitous Computing* 2 (4): 263–277.

Böhmer, Matthias, Christian Lander, Sven Gehring, Duncan P. Brumby, and Antonio Krüger. 2014. "Interrupted by a Phone Call: Exploring Designs for Lowering the Impact of Call Notifications for Smartphone Users". In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems,* 3045–3054. ACM.

Borg, Ingwer, and Thomas Staufenbiel. 2007. *Lehrbuch Theorien und Methoden der Skalierung.* Huber.

Card, Stuart K., Thomas P. Moran, and Allen Newell. 1980. "The Keystroke–Level Model for User Performance Time with Interactive Systems". *Communications of the ACM* 23 (7): 396–410.

Card, Stuart K., Thomas P. Moran, and Allen Newell. 1986. "The Model Human Processor: An Engineering Model of Human Performance". *Handbook of Human Perception* 2.

Cellier, Jean-Marie, and Hélène Eyrolle. 1992. "Interference between switched tasks". *Ergonomics* 35 (1): 25–36.

Chang, Edward Y. 2013. "Context–Aware Computing: Opportunities and Open Issues". *Proceedings of the VLDB Endowment* 6 (11): 1172–1173.

Chen, Guanling, and David Kotz. 2000. *A Survey of Context–Aware Mobile Computing Research.* Technical report. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College.

Chen, Harry, Tim Finin, and Anupam Joshi. 2003. "An Ontology for Context–Aware Pervasive Computing Environments". *The Knowledge Engineering Review* 18 (03): 197–207.

Coiera, Enrico. 2012. "The science of interruption". *BMJ quality & safety* 21 (5): 357–360.

Cutrell, Edward, Mary Czerwinski, and Eric Horvitz. 2001. "Notification, Disruption, and Memory: Effects of Messaging Interruptions on Memory and Performance":263–269.

Czerwinski, Mary, Eric Horvitz, and Susan Wilhite. 2004. "A Diary Study of Task Switching and Interruptions". In *Proceedings of the SIGCHI conference on Human factors in computing systems,* 175–182. ACM.

Dabbish, Laura, Gloria Mark, and Víctor M González. 2011. "Why Do I Keep Interrupting Myself?: Environment, Habit and Self–Interruption". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 3127–3130. ACM.

Dayer, Lindsey, Seth Heldenbrand, Paul Anderson, Paul O. Gubbins, and Bradley C. Martin. 2013. "Smartphone Medication Adherence Apps: Potential Benefits to Patients and Providers". *Journal of the American Pharmacists Association: JAPhA* 53 (2).

Dey, Anind K., and Gregory D. Abowd. 2000. "CybreMinder: A Context–Aware System for Supporting Reminders". In *Proceedings of the 2nd International Symposium on Handheld and Ubiquitous Computing,* 172–186. HUC '00. Bristol, UK: Springer. ISBN: 3-540-41093-7.

Ferreira, Denzil, Jorge Gonçalves, Vassilis Kostakos, Louise Barkhuus, and Anind K Dey. 2014. "Contextual Experience Sampling of Mobile Application Micro–Usage".

Foushee, H. Clayton, and Robert L. Helmreich. 1988. "Group Interaction and Flight Crew Performance". *Human factors in aviation:*189.

Franke, Jerry L., Jody J. Daniels, and Daniel C. McFarlane. 2002. "Recovering Context after Interruption". In *Proceedings 24th Annual Meeting of the Cognitive Science Society (CogSci 2002),* 310–315.

Gillie, Tony, and Donald Broadbent. 1989. "What makes interruptions disruptive? A study of length, similarity, and complexity". *Psychological Research* 50 (4): 243–250.

Godbole, A., and W.W. Smari. 2006. "A Methodology and Design Process for System Generated User Interruption based on Context, Preferences, and Situation Awareness". In *Information Reuse and Integration, 2006 IEEE International Conference on,* 608–616.

Harr, Rikard, and Victor Kaptelinin. 2007. "Unpacking the Social Dimension of External Interruptions". In *Proceedings of the 2007 international ACM conference on Supporting group work,* 399–408. ACM.

Harr, Rikard, and Mikael Wiberg. 2008. "Lost in translation: investigating the ambiguity of availability cues in an online media space". *Behaviour & Information Technology* 27 (3): 243–262.

Henricksen, Karen, and Jadwiga Indulska. 2006. "Developing Context–Aware Pervasive Computing Applications: Models and Approach". *Pervasive and mobile computing* 2 (1): 37–64.

Ho, Joyce, and Stephen S. Intille. 2005. "Using Context–Aware Computing to Reduce the Perceived Burden of Interruptions from Mobile Devices". In *Proceedings of the SIGCHI conference on Human factors in computing systems,* 909–918. ACM.

Holleis, Paul, Friederike Otto, Heinrich Hussmann, and Albrecht Schmidt. 2007. "Keystroke–Level Model for Advanced Mobile Phone Interaction". In *Proceedings of the SIGCHI conference on Human factors in computing systems,* 1505–1514. ACM.

Horvitz, Eric, and Johnson Apacible. 2003. "Learning and Reasoning about Interruption". In *Proceedings of the 5th international conference on Multimodal interfaces,* 20–27. ACM.

Horvitz, Eric, Andy Jacobs, and David Hovel. 1999. "Attention–Sensitive Alerting". In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence,* 305–313. Morgan Kaufmann organizations Inc.

Horvitz, Eric, Carl Kadie, Tim Paek, and David Hovel. 2003. "Models of Attention in Computing and Communication: From Principles to Applications". *Communications of the ACM* 46 (3): 52–59.

Horvitz, Eric, Paul Koch, and Johnson Apacible. 2004. "BusyBody: Creating and Fielding Personalized Models of the Cost of Interruption". In *Proceedings of the 2004 ACM conference on Computer supported cooperative work,* 507–510. ACM.

Horvitz, Eric, Paul Koch, Carl M. Kadie, and Andy Jacobs. 2002. "Coordinate: Probabilistic forecasting of presence and availability". In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence,* 224–233. Morgan Kaufmann organizations Inc.

Hudson, James M., Jim Christensen, Wendy A. Kellogg, and Thomas Erickson. 2002. "I'd Be Overwhelmed, But It's Just One More Thing to Do: Availability and Interruption in Research Management". In *Proceedings of the SIGCHI Conference on Human factors in computing systems,* 97–104. ACM.

Jastrzembski, Tiffany S., and Neil Charness. 2007. "The Model Human Processor and the older adult: parameter estimation and validation within a mobile phone task." *Journal of experimental psychology: applied* 13 (4): 224.

John, Bonnie E., and Wayne D. Gray. 1995. "CPM-GOMS: An Analysis Method for Tasks with Parallel Activities". In *Conference companion on Human factors in computing systems,* 393–394. ACM.

John, Bonnie E., and David E. Kieras. 1996. "The GOMS Family of User Interface Analysis Techniques: Comparison and Contrast". *ACM Transactions on Computer-Human Interaction (TOCHI)* 3 (4): 320–351.

Kamar, Ece, and Eric Horvitz. 2010. "Investigation of Principles of Context–Sensitive Reminding".

Kamar, Ece, and Eric Horvitz. 2011. "Jogger: Models for Context–Sensitive Reminding". In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 3,* 1089–1090. AAMAS '11. Taipei, Taiwan: International Foundation for Autonomous Agents and Multiagent Systems. ISBN: 0-9826571-7-X, 978-0-9826571-7-1.

Kumar, Anup, and Bin Xie. 2012. *Handbook of Mobile Systems Applications and Services.* Volume 1. CRC Press.

Kwapisz, Jennifer R., Gary M. Weiss, and Samuel A. Moore. 2011. "Activity recognition using cell phone accelerometers". *ACM SigKDD Explorations Newsletter* 12 (2): 74–82.

Latorella, Kara A. 1999. *Investigating Interruptions: Implications for Flightdeck Performance.* Volume 99. 209707. National Aeronautics / Space Administration, Langley Research Center.

Leiva, Luis, Matthias Böhmer, Sven Gehring, and Antonio Krüger. 2012. "Back to the App: The Costs of Mobile Application Interruptions". In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services,* 291–294. ACM.

Marmasse, Natalia, and Chris Schmandt. 2000. "Location–aware information delivery with comMotion". In *Handheld and Ubiquitous Computing,* 157–171. Springer.

McFarlane, Daniel C. 1997. *Interruption of People in Human-Computer Interaction: A General Unifying Definition of Human Interruption and Taxonomy.* Technical report. DTIC Document.

McFarlane, Daniel C. 2002. "Comparison of Four Primary Methods for Coordinating the Interruption of People in Human-Computer Interaction". *Human-Computer Interaction* 17 (1): 63–139.

McFarlane, Daniel C., and Kara A Latorella. 2002. "The Scope and Importance of Human Interruption in Human–Computer Interaction Design". *Human-Computer Interaction* 17 (1): 1–61.

Miller, George A. 1956. "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information". *Psychological review* 63 (2): 81.

Miller, George A., Eugene Galanter, and Karl H. Pribram. 1986. *Plans and the Structure of Behavior.* Adams Bannister Cox.

Nicklas, Daniela, Matthias Grossmann, Jorge Minguez, and Matthias Wieland. 2008. "Adding High–level Reasoning to Efficient Low–level Context Management: A Hybrid Approach". In *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on,* 447–452. IEEE.

O'Conaill, Brid, and David Frohlich. 1995. "Timespace in the Workplace: Dealing with Interruptions". In *Conference companion on Human factors in computing systems,* 262–263. ACM.

Oulasvirta, Antti, Sakari Tamminen, Virpi Roto, and Jaana Kuorelahti. 2005. "Interaction in 4–second bursts: the fragmented nature of attentional resources in mobile HCI". In *Proceedings of the SIGCHI conference on Human factors in computing systems,* 919–928. ACM.

Pavel, Misha, Holly Jimison, Tamara Hayes, Nicole Larimer, Stuart Hagler, Yves Vimegnon, Todd Leen, and Umut Ozertem. 2010. "Optimizing Medication Reminders Using a Decision–Theoretic Framework". *Stud Health Technol Inform* 160 (pt 2): 791–795.

Pejovic, Veljko, and Mirco Musolesi. 2014. "Anticipatory Mobile Computing for Behaviour Change Interventions". In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication,* 1025–1034. ACM.

Pejovic, Veljko, and Mirco Musolesi. 2014. "InterruptMe: Designing Intelligent Prompting Mechanisms for Pervasive Applications". UbiComp.

Perera, Charith, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. 2014. "Context Aware Computing for The Internet of Things: A Survey". *Communications Surveys & Tutorials, IEEE* 16 (1): 414–454.

Pielot, Martin. 2014. "Large–Scale Evaluation of Call–Availability Prediction". In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing,* 933–937. ACM.

Pielot, Martin, Karen Church, and Rodrigo de Oliveira. 2014. "An In–Situ Study of Mobile Phone Notifications". In *Proc. MobileHCI,* volume 14.

Poppinga, B., W. Heuten, and S. Boll. 2014. "Sensor–Based Identification of Opportune Moments for Triggering Notifications". *Pervasive Computing, IEEE* 13, number 1 (): 22–29. ISSN: 1536-1268.

Preece, Jenny, Yvonne Rogers, Helen Sharp, David Benyon, Simon Holland, and Tom Carey. 1994. *Human–Computer Interaction.* Addison–Wesley Longman Ltd.

Schilit, Bill, Norman Adams, and Roy Want. 1994. "Context–Aware Computing Applications". In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on,* 85–90. IEEE.

Vervloet, Marcia, Annemiek J. Linn, Julia C. M. van Weert, Dinny H. de Bakker, Marcel L. Bouvy, and Liset van Dijk. 2012. "The effectiveness of interventions using electronic reminders to improve adherence to chronic medication: a systematic review of the literature". *JAMIA* 19 (5): 696–704.

Want, Roy, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. 1992. "The Active Badge Location System". *ACM Transactions on Information Systems (TOIS)* 10 (1): 91–102.

Yan, Tingxin, David Chu, Deepak Ganesan, Aman Kansal, and Jie Liu. 2012. "Fast App Launching for Mobile Devices Using Predictive User Context". In *Proceedings of the 10th international conference on Mobile systems, applications, and services,* 113–126. ACM.

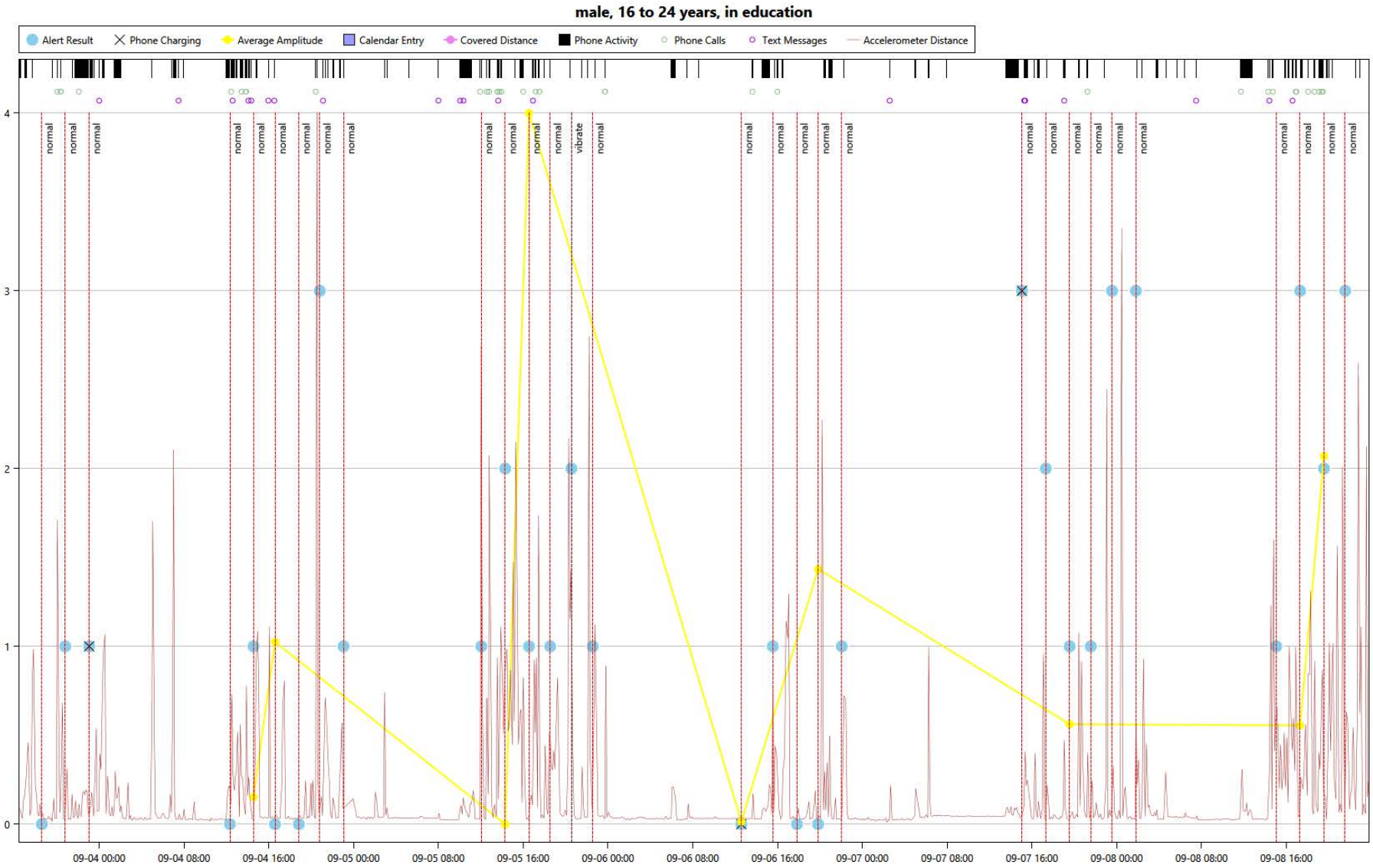
Zeigarnik, Bluma. 1927. "Über das Behalten von erledigten und unerledigten Handlungen". *Psychologische Forschung* 9 (1): 1–85.

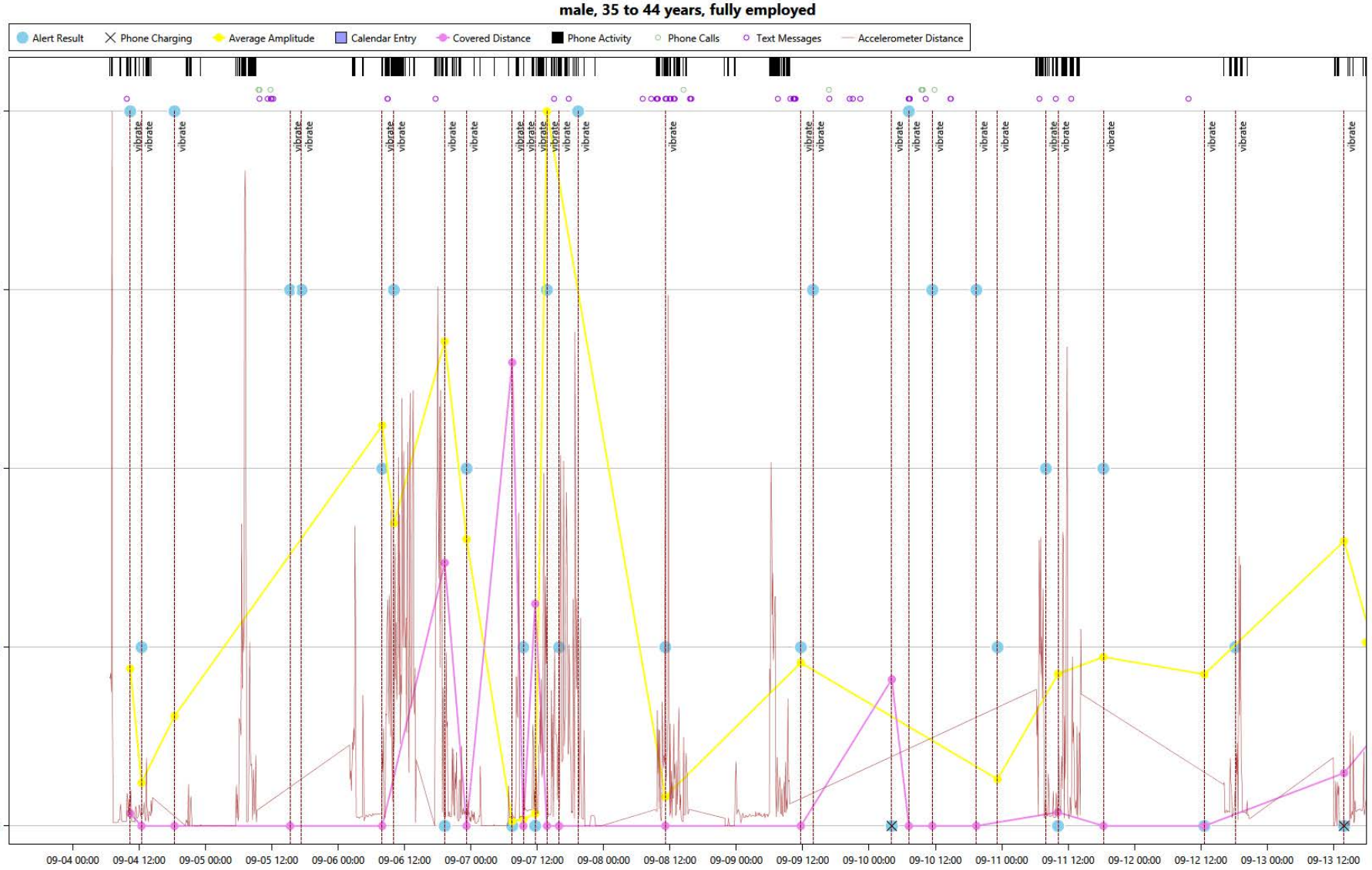
Zulkernain, Sina, Praveen Madiraju, and Sheikh Iqbal Ahamed. 2010. "A Context Aware Interruption Management System for Mobile Devices". In *Mobile Wireless Middleware, Operating Systems, and Applications,* 221–234.

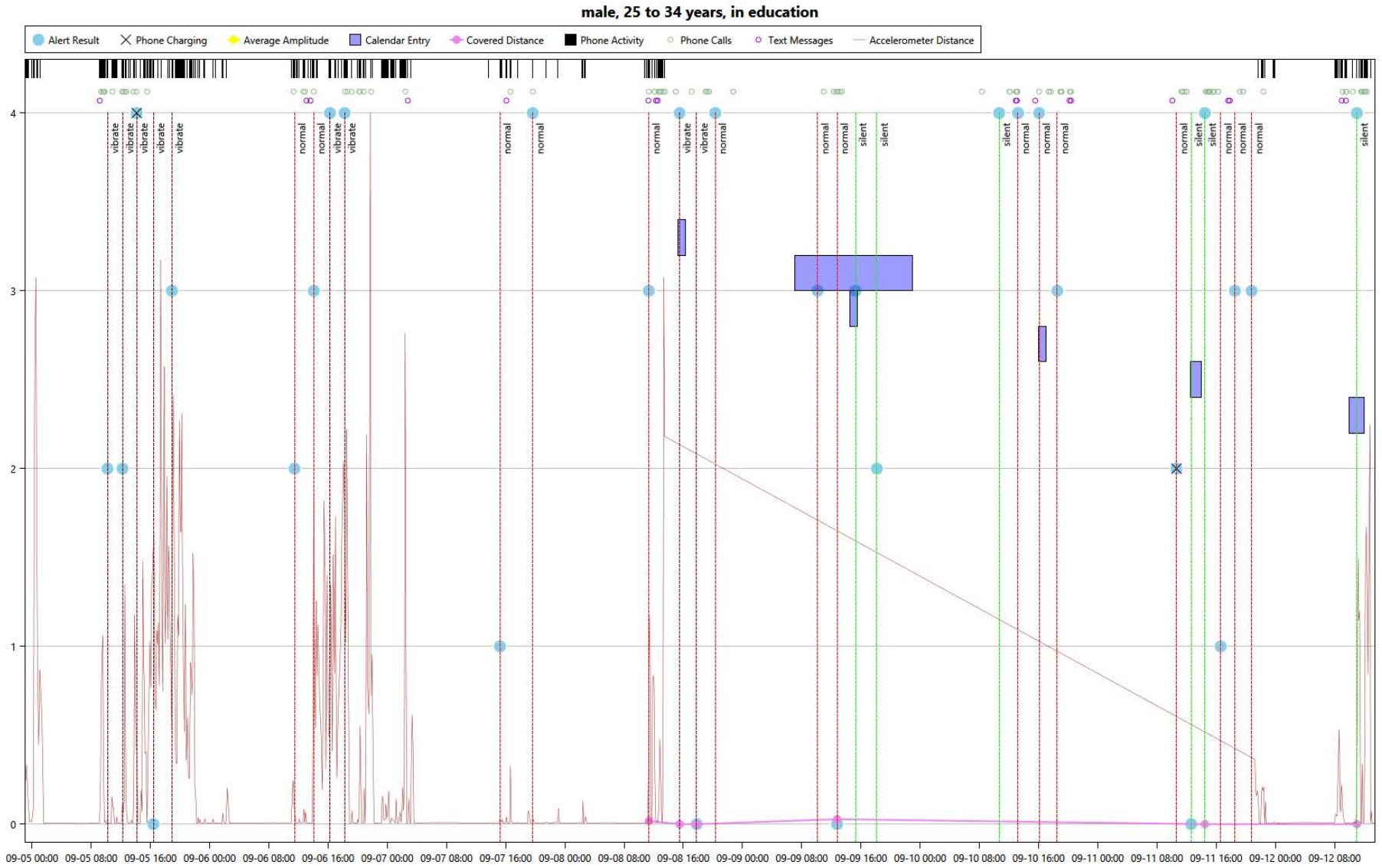
Zulkernain, Sina, Praveen Madiraju, and Sheikh Iqbal Ahamed. 2011. "A Context–aware Cost of Interruption Model for Mobile Devices". In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on,* 456–460. IEEE.

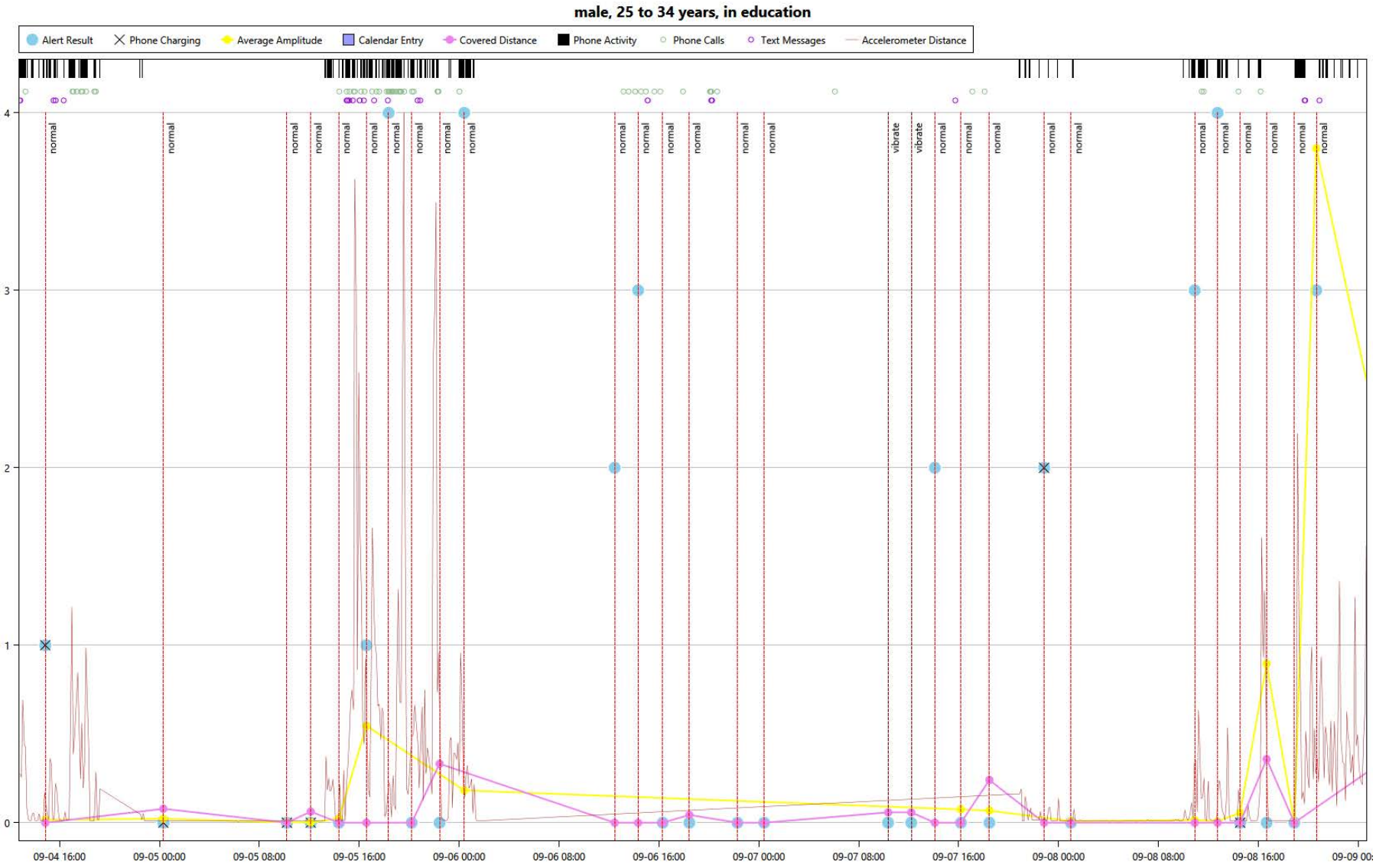
A Test Results

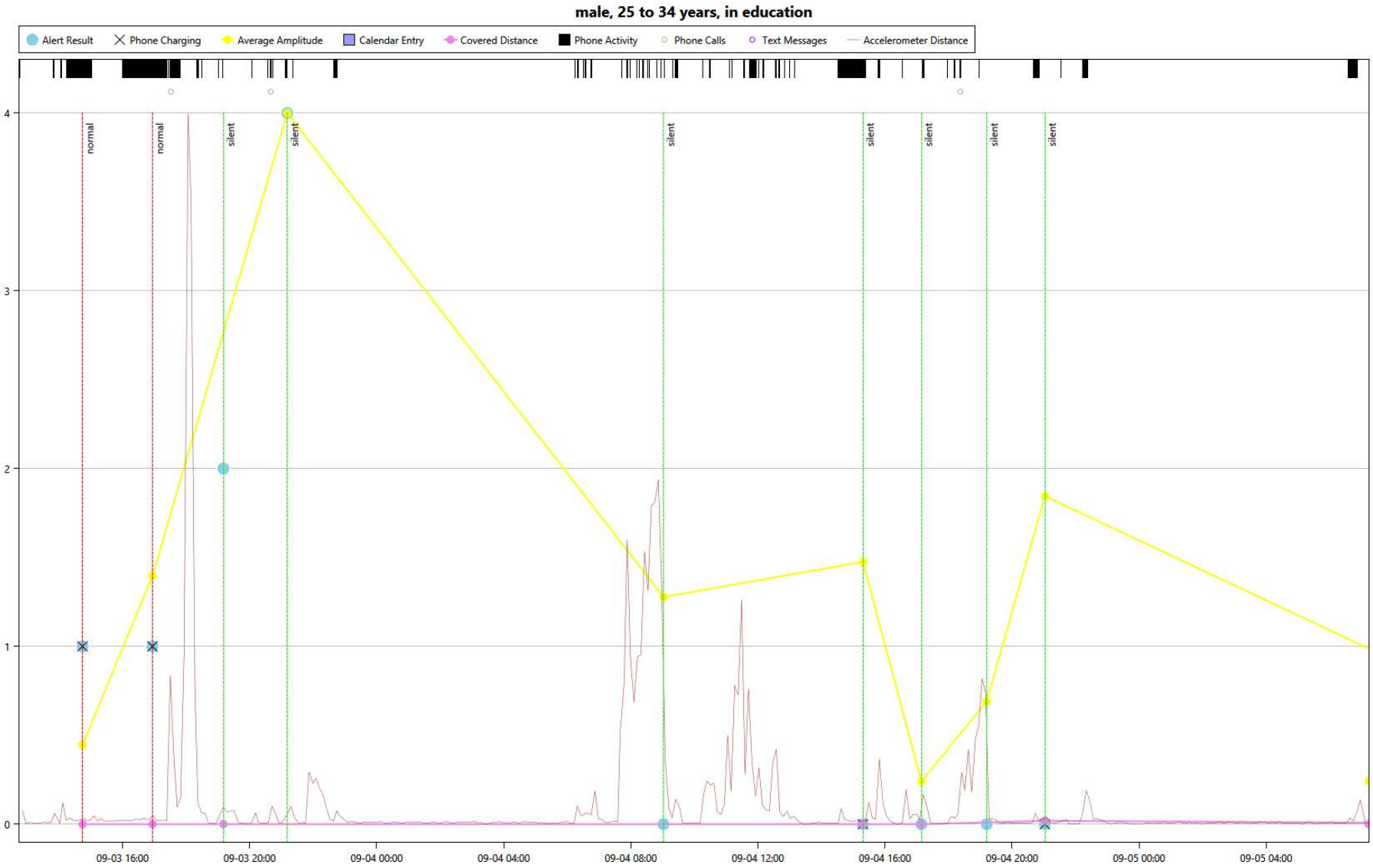
Here are the 2D plots for each of our 32 participants which have been used for the evaluation of our study results in Chapter 4.3. The x–axis expresses the study length and the y–axis is assigned to alert results (zero for missed alerts and one to four for the individual results). Y–axis values of amplitude and accelerometer data are normalized to the overall maximum value of one individual user, whereas the covered distance is normalized to the overall maximum value of all users.

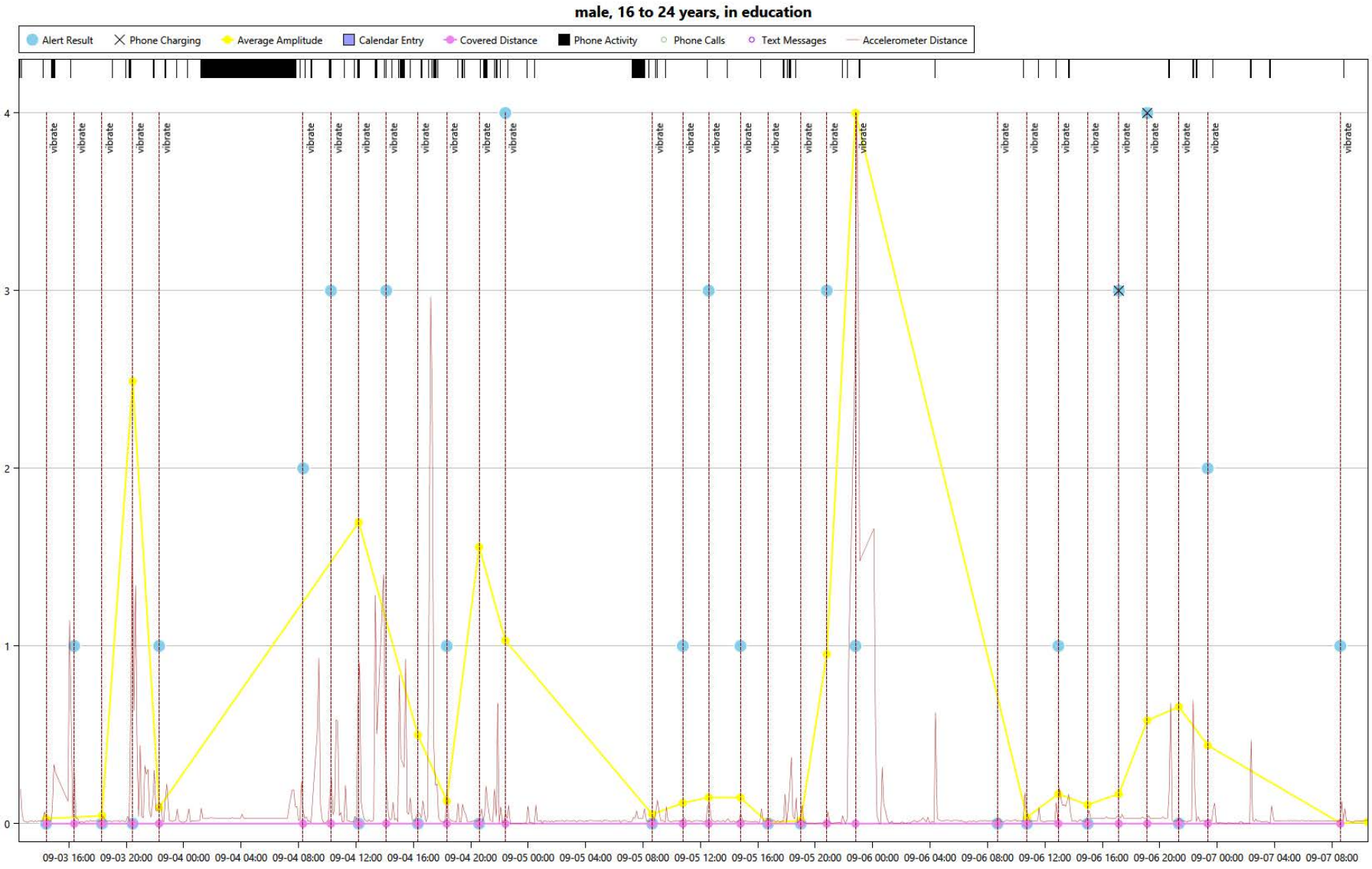

male, 16 to 24 years, in education
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
4
3
2
1
0
normal
vibrate
09-04 00:00
09-04 08:00
09-04 16:00
09-05 00:00
09-05 08:00
09-05 16:00
09-06 00:00
09-06 08:00
09-06 16:00
09-07 00:00
09-07 08:00
09-07 16:00
09-08 00:00
09-08 08:00
09-08 16:00

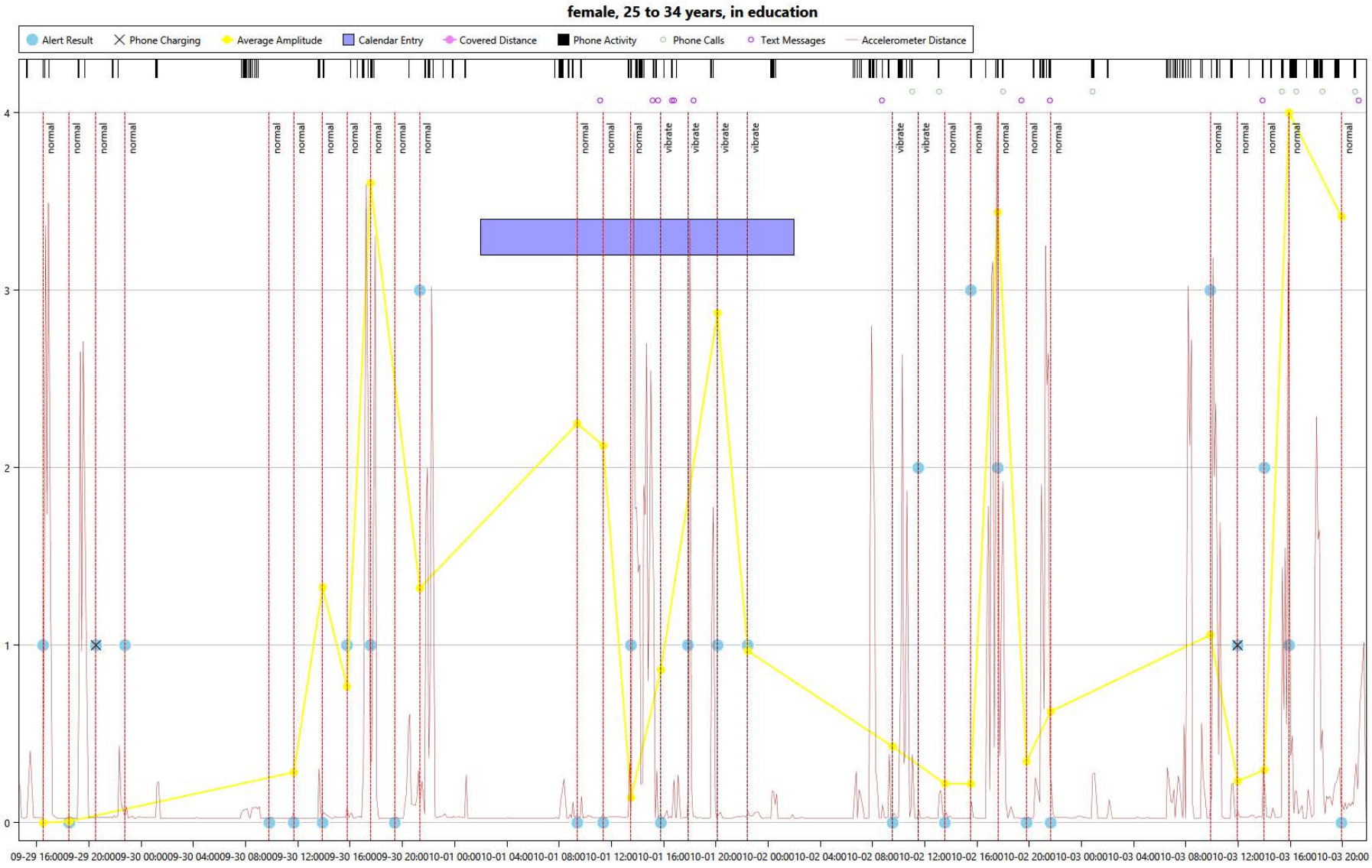

male, 35 to 44 years, fully employed
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
vibrate
4
3
2
1
0
09-04 00:00
09-04 12:00
09-05 00:00
09-05 12:00
09-06 00:00
09-06 12:00
09-07 00:00
09-07 12:00
09-08 00:00
09-08 12:00
09-09 00:00
09-09 12:00
09-10 00:00
09-10 12:00
09-11 00:00
09-11 12:00
09-12 00:00
09-12 12:00
09-13 00:00
09-13 12:00

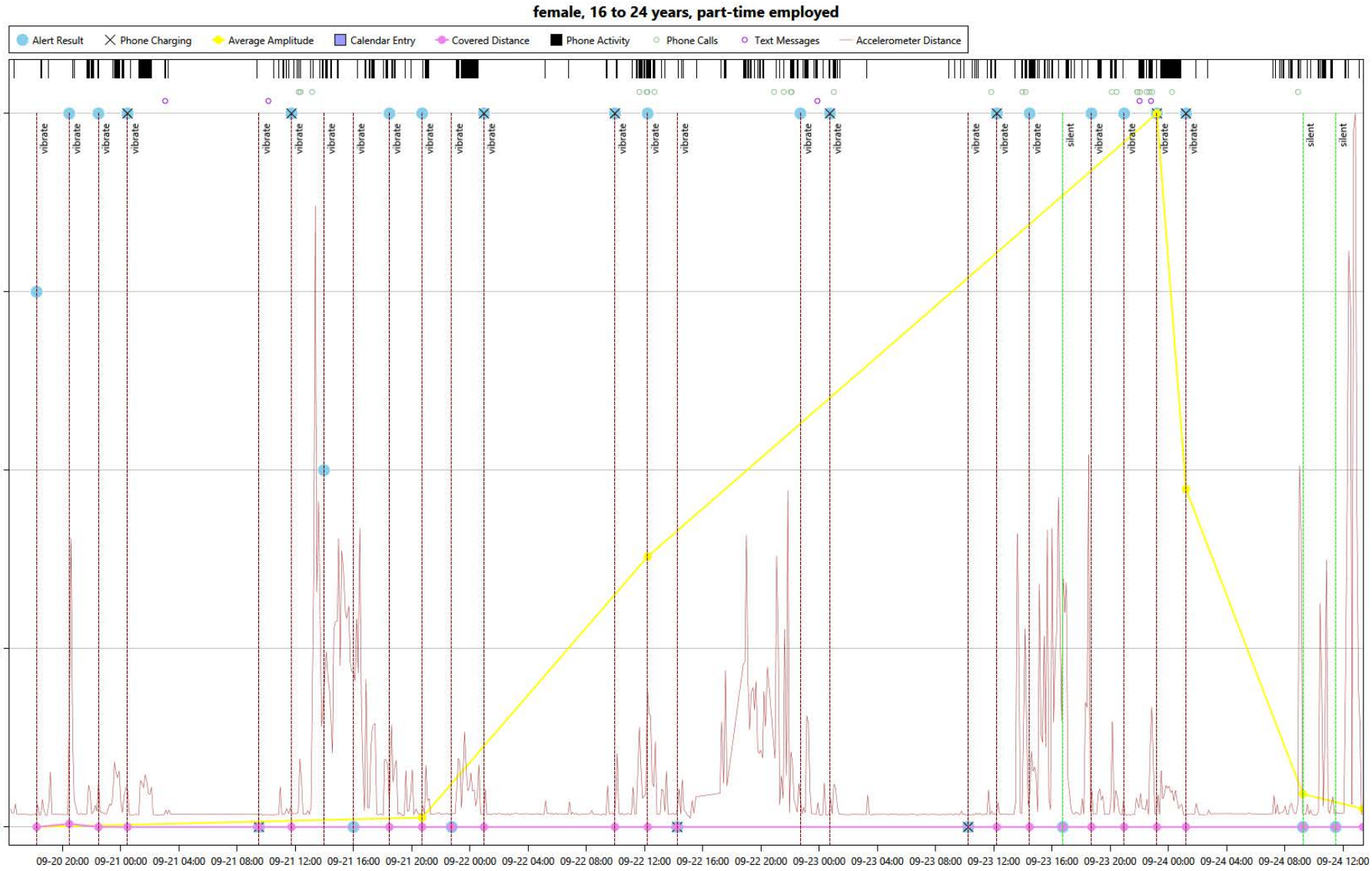

male, 25 to 34 years, in education
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
4
3
2
1
0
vibrate
normal
silent
09-05 00:00
09-05 08:00
09-05 16:00
09-06 00:00
09-06 08:00
09-06 16:00
09-07 00:00
09-07 08:00
09-07 16:00
09-08 00:00
09-08 08:00
09-08 16:00
09-09 00:00
09-09 08:00
09-09 16:00
09-10 00:00
09-10 08:00
09-10 16:00
09-11 00:00
09-11 08:00
09-11 16:00
09-12 00:00
09-12 08:00

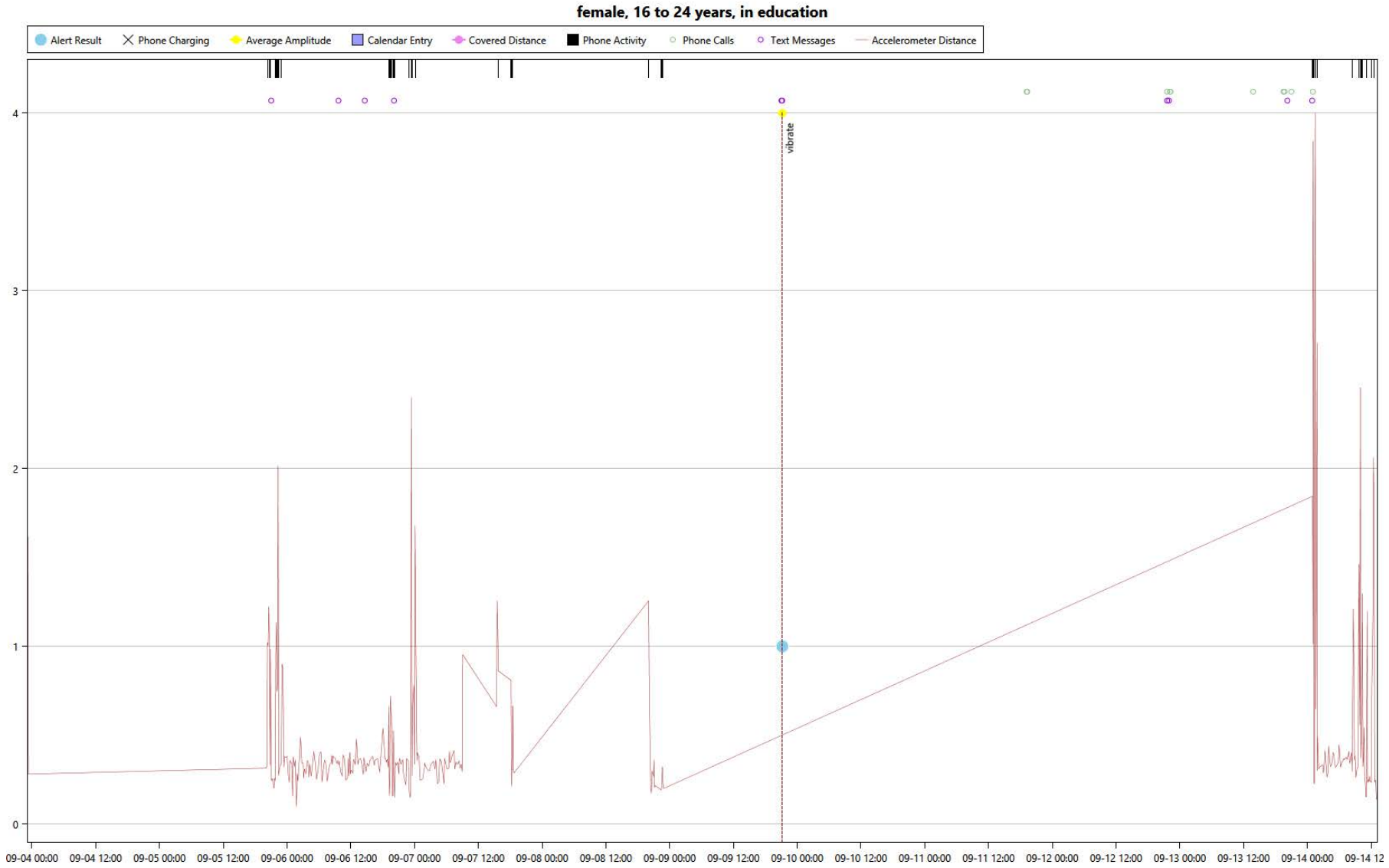

male, 25 to 34 years, in education
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
4
3
2
1
0
normal
vibrate
09-04 16:00
09-05 00:00
09-05 08:00
09-05 16:00
09-06 00:00
09-06 08:00
09-06 16:00
09-07 00:00
09-07 08:00
09-07 16:00
09-08 00:00
09-08 08:00
09-08 16:00
09-09 00:

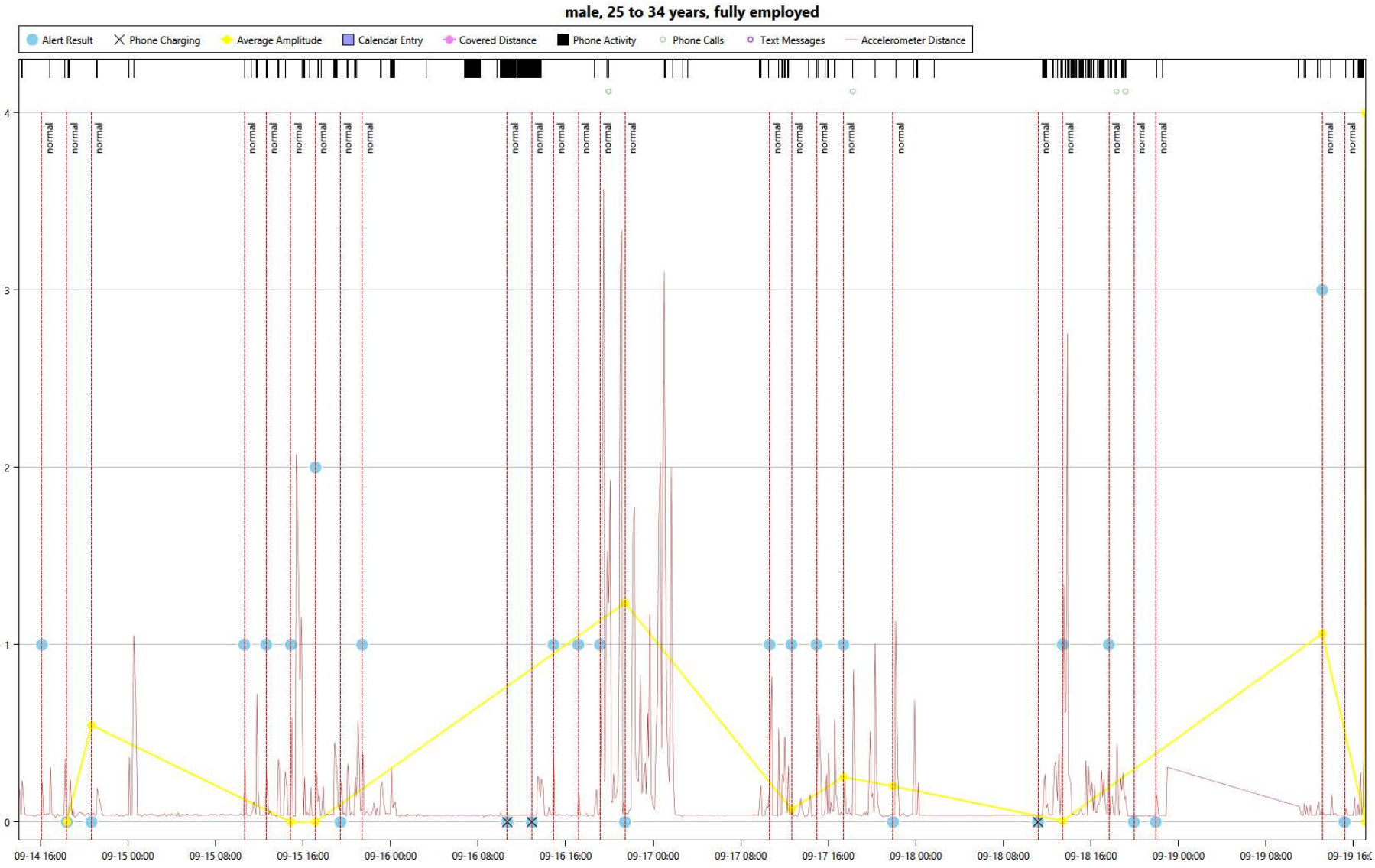

male, 25 to 34 years, in education
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
normal
normal
silent
silent
silent
silent
silent
silent
silent
4
3
2
1
0
09-03 16:00
09-03 20:00
09-04 00:00
09-04 04:00
09-04 08:00
09-04 12:00
09-04 16:00
09-04 20:00
09-05 00:00
09-05 04:00

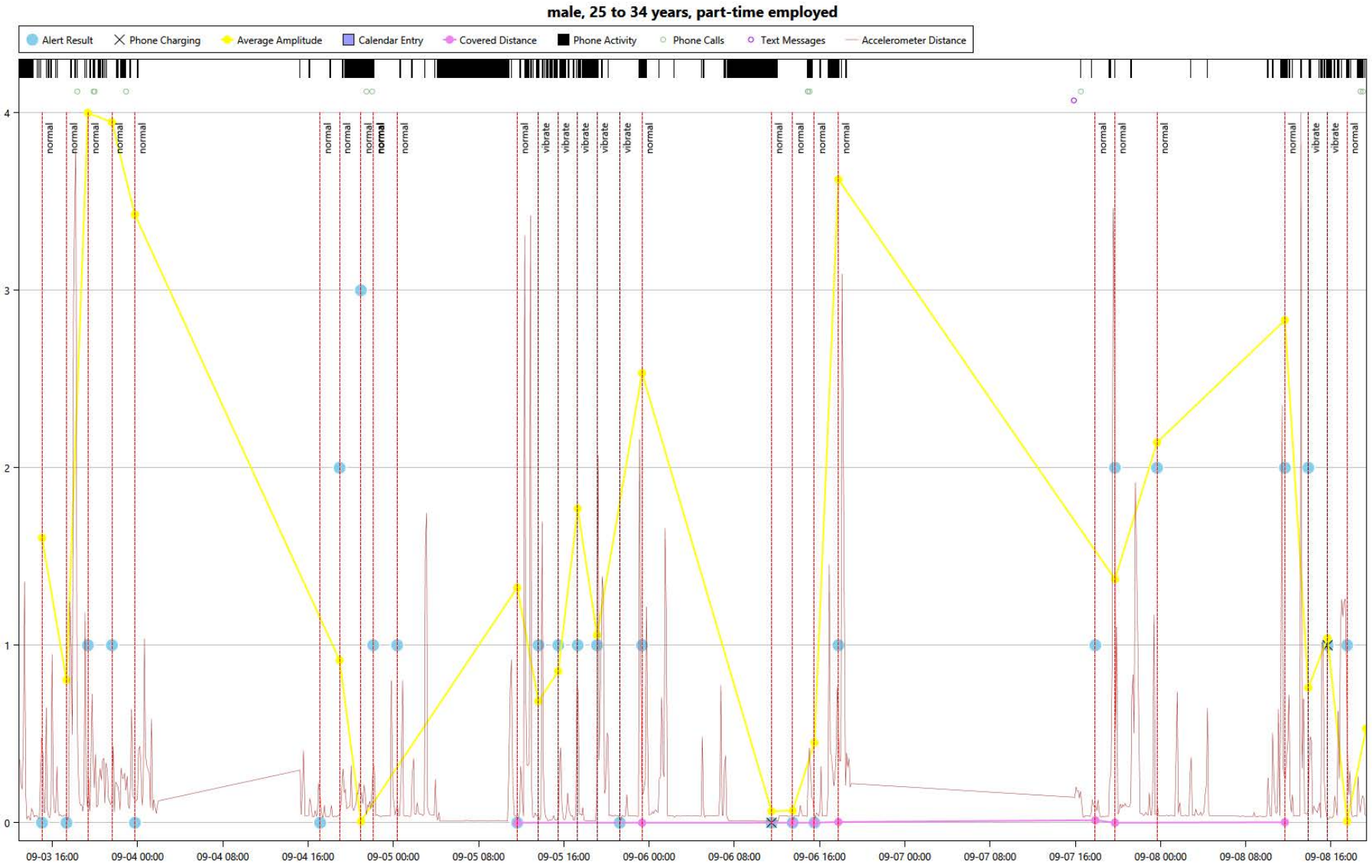

male, 16 to 24 years, in education
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
vibrate
4
3
2
1
0
09-03 16:00
09-03 20:00
09-04 00:00
09-04 04:00
09-04 08:00
09-04 12:00
09-04 16:00
09-04 20:00
09-05 00:00
09-05 04:00
09-05 08:00
09-05 12:00
09-05 16:00
09-05 20:00
09-06 00:00
09-06 04:00
09-06 08:00
09-06 12:00
09-06 16:00
09-06 20:00
09-07 00:00
09-07 04:00
09-07 08:00

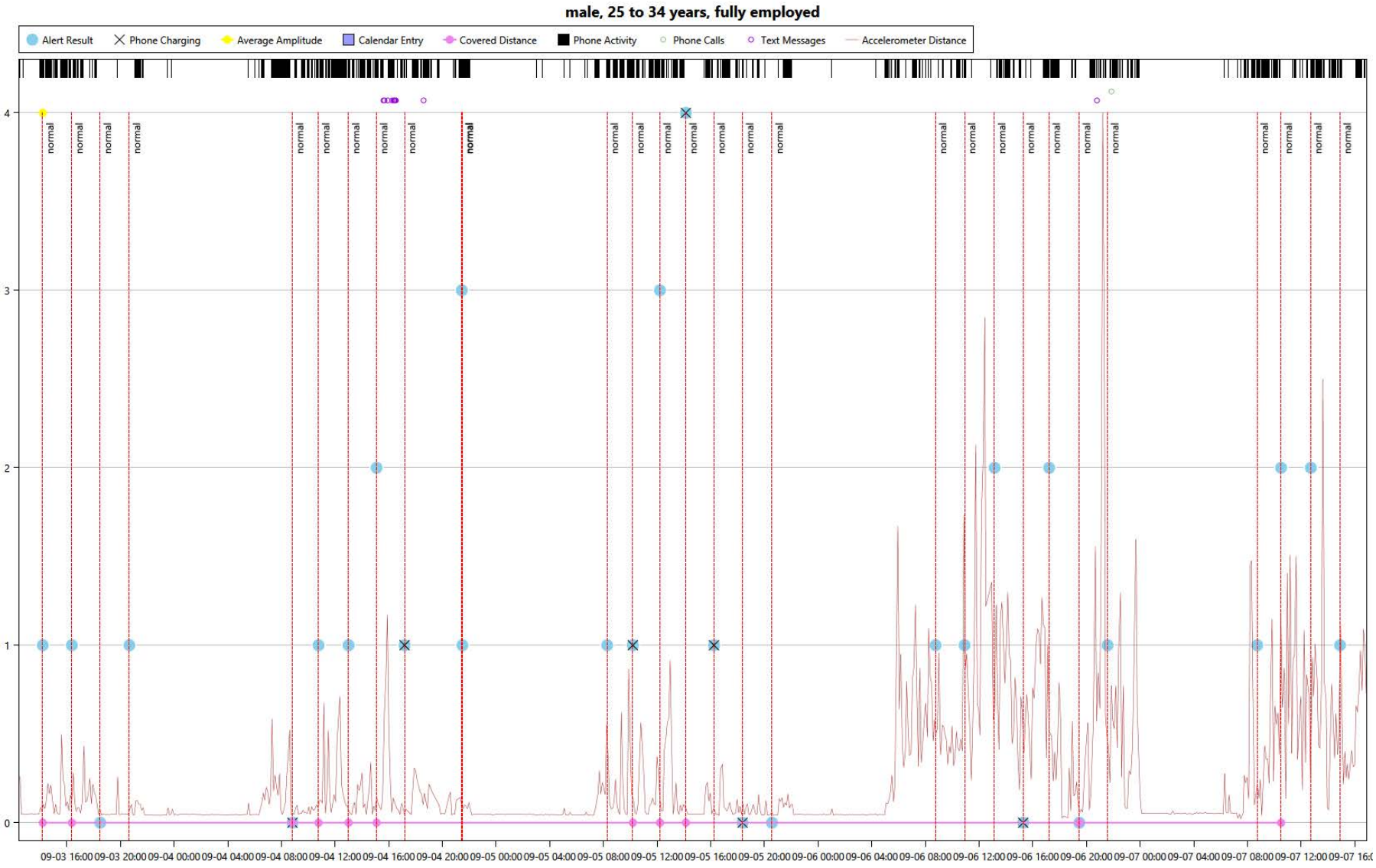

female, 25 to 34 years, in education
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
4
3
2
1
0
normal
vibrate
09-29 16:00
09-29 20:00
09-30 00:00
09-30 04:00
09-30 08:00
09-30 12:00
09-30 16:00
09-30 20:00
10-01 00:00
10-01 04:00
10-01 08:00
10-01 12:00
10-01 16:00
10-01 20:00
10-02 00:00
10-02 04:00
10-02 08:00
10-02 12:00
10-02 16:00
10-02 20:00
10-03 00:00
10-03 04:00
10-03 08:00
10-03 12:00
10-03 16:00
10-03 20:00


female, 16 to 24 years, part-time employed
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
4
3
2
1
0
vibrate
silent
09-20 20:00
09-21 00:00
09-21 04:00
09-21 08:00
09-21 12:00
09-21 16:00
09-21 20:00
09-22 00:00
09-22 04:00
09-22 08:00
09-22 12:00
09-22 16:00
09-22 20:00
09-23 00:00
09-23 04:00
09-23 08:00
09-23 12:00
09-23 16:00
09-23 20:00
09-24 00:00
09-24 04:00
09-24 08:00
09-24 12:00

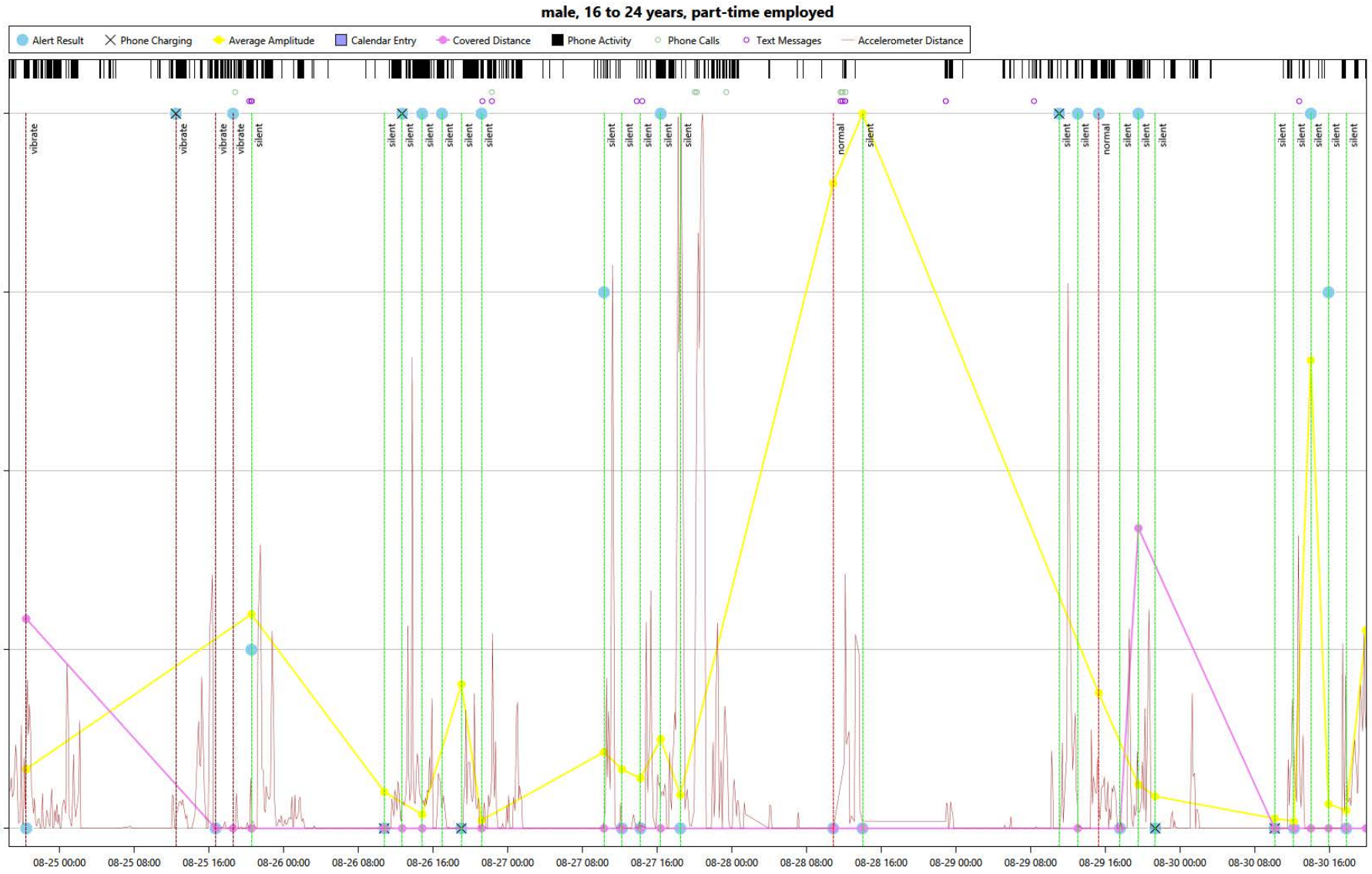

female, 16 to 24 years, in education
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
vibrate
4
3
2
1
0
09-04 00:00
09-04 12:00
09-05 00:00
09-05 12:00
09-06 00:00
09-06 12:00
09-07 00:00
09-07 12:00
09-08 00:00
09-08 12:00
09-09 00:00
09-09 12:00
09-10 00:00
09-10 12:00
09-11 00:00
09-11 12:00
09-12 00:00
09-12 12:00
09-13 00:00
09-13 12:00
09-14 00:00
09-14 12

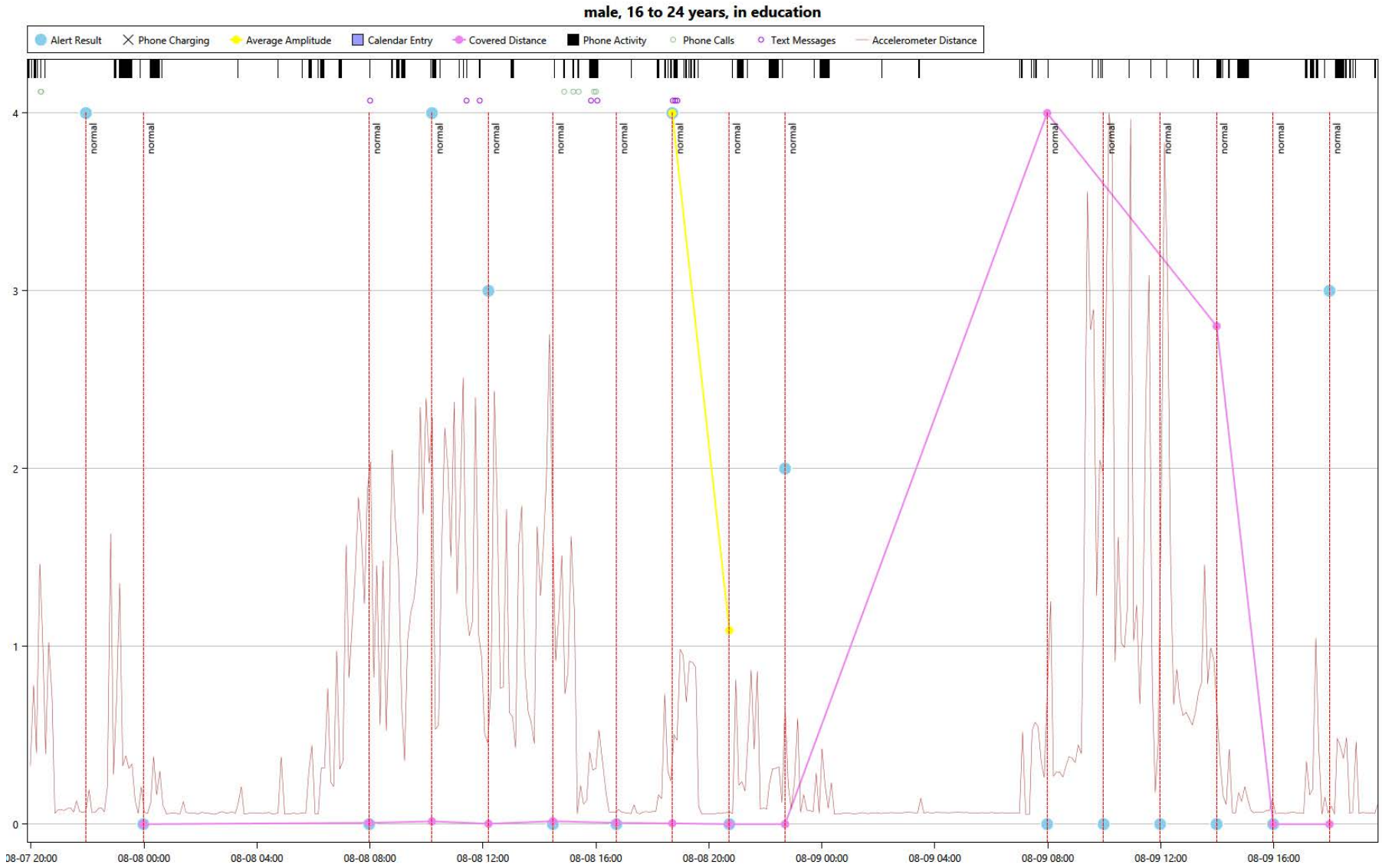

male, 25 to 34 years, fully employed
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
normal
4
3
2
1
0
09-14 16:00
09-15 00:00
09-15 08:00
09-15 16:00
09-16 00:00
09-16 08:00
09-16 16:00
09-17 00:00
09-17 08:00
09-17 16:00
09-18 00:00
09-18 08:00
09-18 16:00
09-19 00:00
09-19 08:00

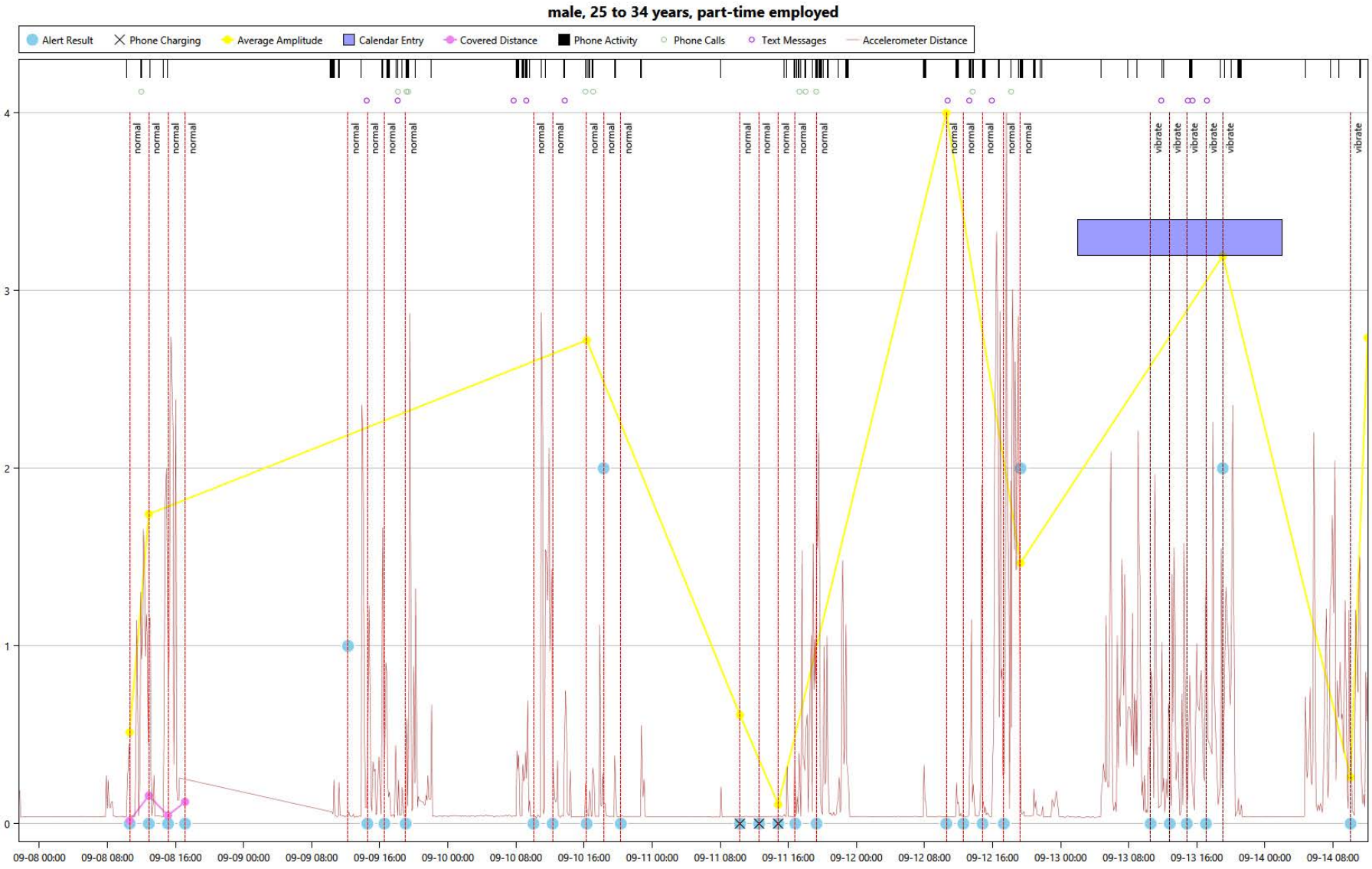

male, 25 to 34 years, part-time employed
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
4
3
2
1
0
normal
vibrate
09-03 16:00
09-04 00:00
09-04 08:00
09-04 16:00
09-05 00:00
09-05 08:00
09-05 16:00
09-06 00:00
09-06 08:00
09-06 16:00
09-07 00:00
09-07 08:00
09-07 16:00
09-08 00:00
09-08 08:00
09-08 16:00

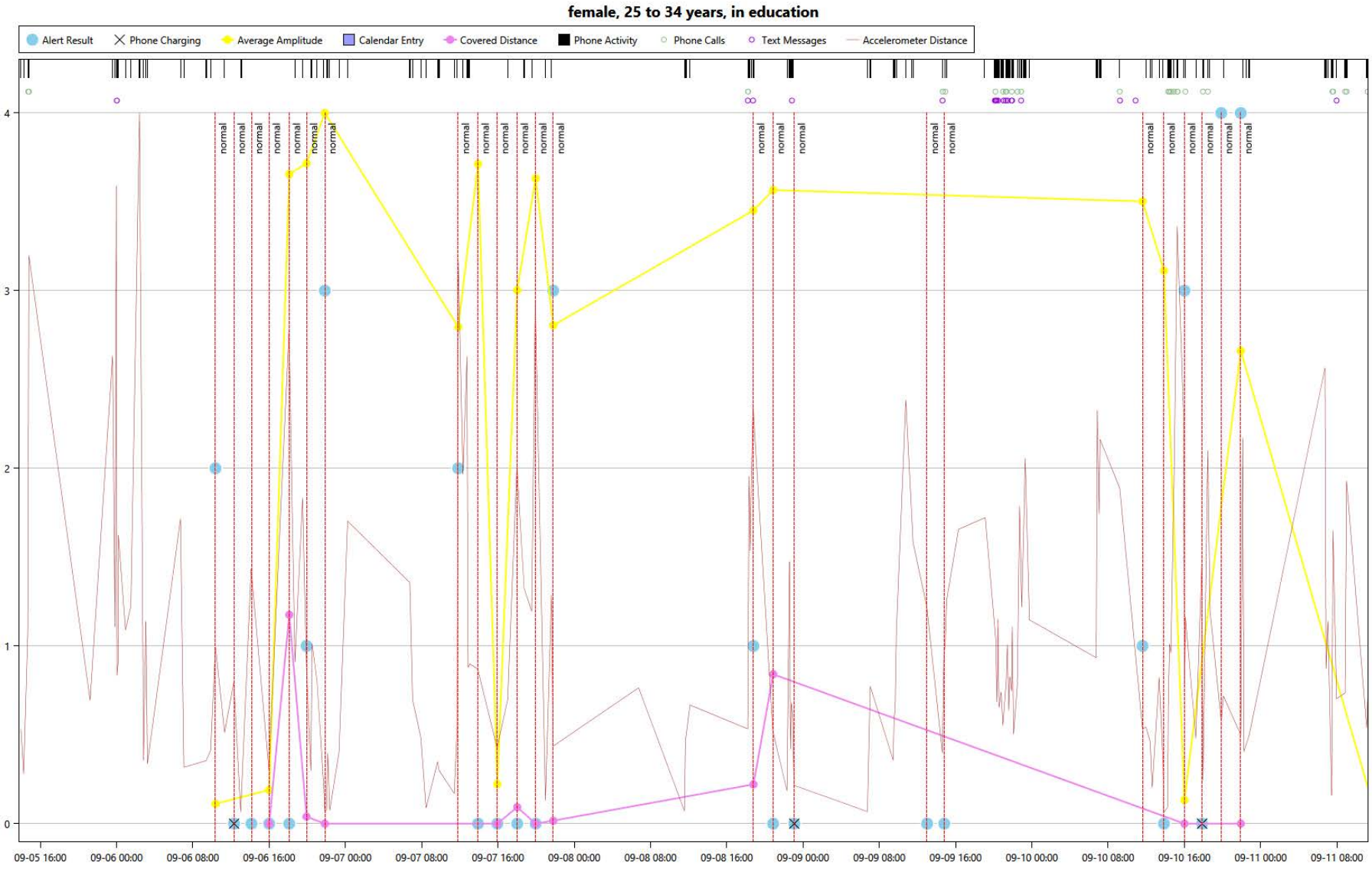

male, 25 to 34 years, fully employed
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
4
3
2
1
0
normal
09-03 16:00
09-03 20:00
09-04 00:00
09-04 04:00
09-04 08:00
09-04 12:00
09-04 16:00
09-04 20:00
09-05 00:00
09-05 04:00
09-05 08:00
09-05 12:00
09-05 16:00
09-05 20:00
09-06 00:00
09-06 04:00
09-06 08:00
09-06 12:00
09-06 16:00
09-06 20:00
09-07 00:00
09-07 04:00
09-07 08:00
09-07 12:00

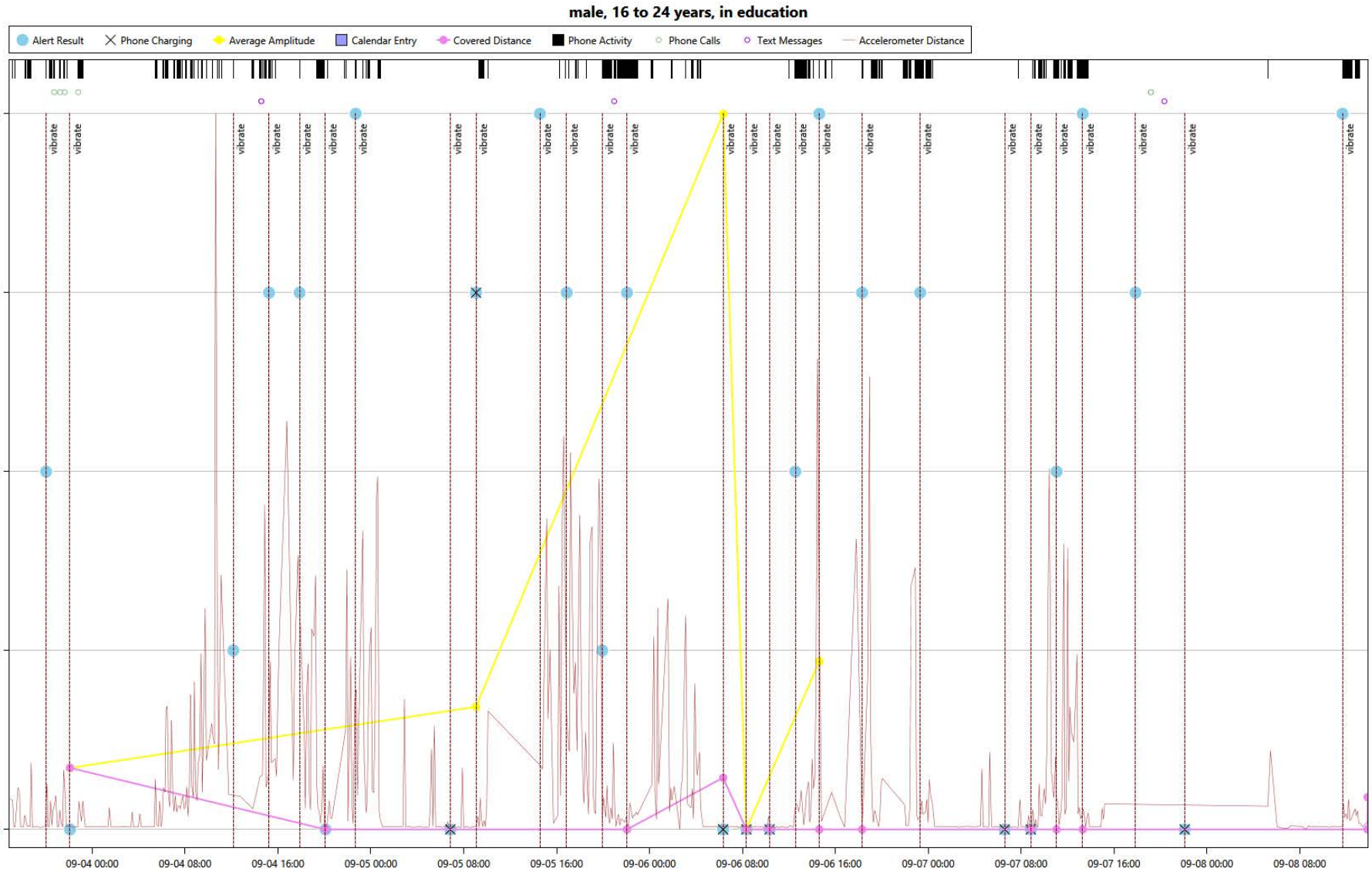

female, 25 to 34 years, in education
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
vibrate
4
3
2
1
0
09-04 16:00
09-05 00:00
09-05 08:00
09-05 16:00
09-06 00:00
09-06 08:00
09-06 16:00
09-07 00:00
09-07 08:00
09-07 16:00
09-08 00:00
09-08 08:00
09-08 16:00
09-09 00:00
09-09 08:00
09-09 16:00
09-10 00:00
09-10 08:00
09-10 16:00
09-11 00:00
09-11 08:00
09-11 16:00

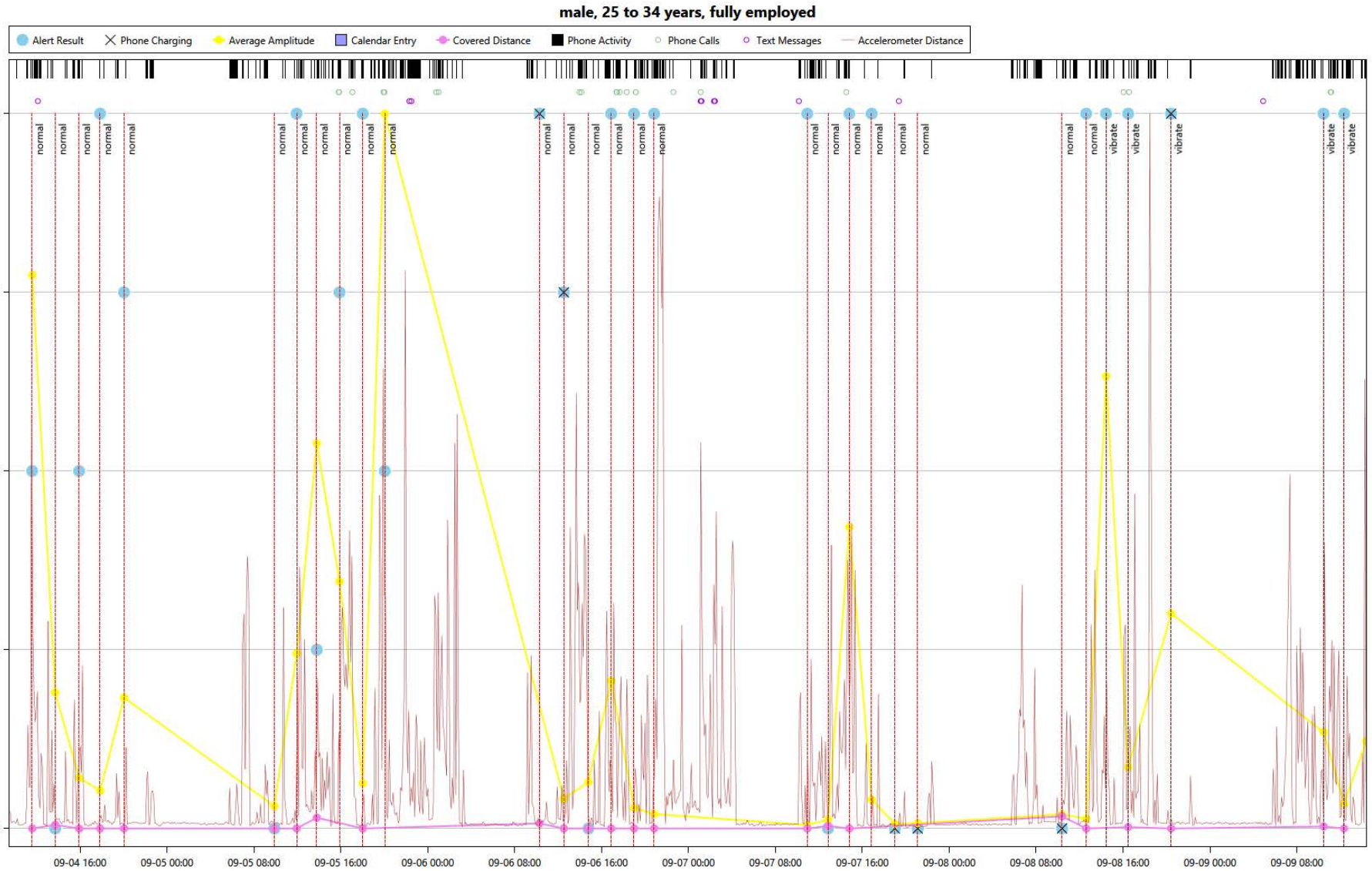

male, 16 to 24 years, part-time employed
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
4
3
2
1
0
vibrate
silent
normal
08-25 00:00
08-25 08:00
08-25 16:00
08-26 00:00
08-26 08:00
08-26 16:00
08-27 00:00
08-27 08:00
08-27 16:00
08-28 00:00
08-28 08:00
08-28 16:00
08-29 00:00
08-29 08:00
08-29 16:00
08-30 00:00
08-30 08:00
08-30 16:00

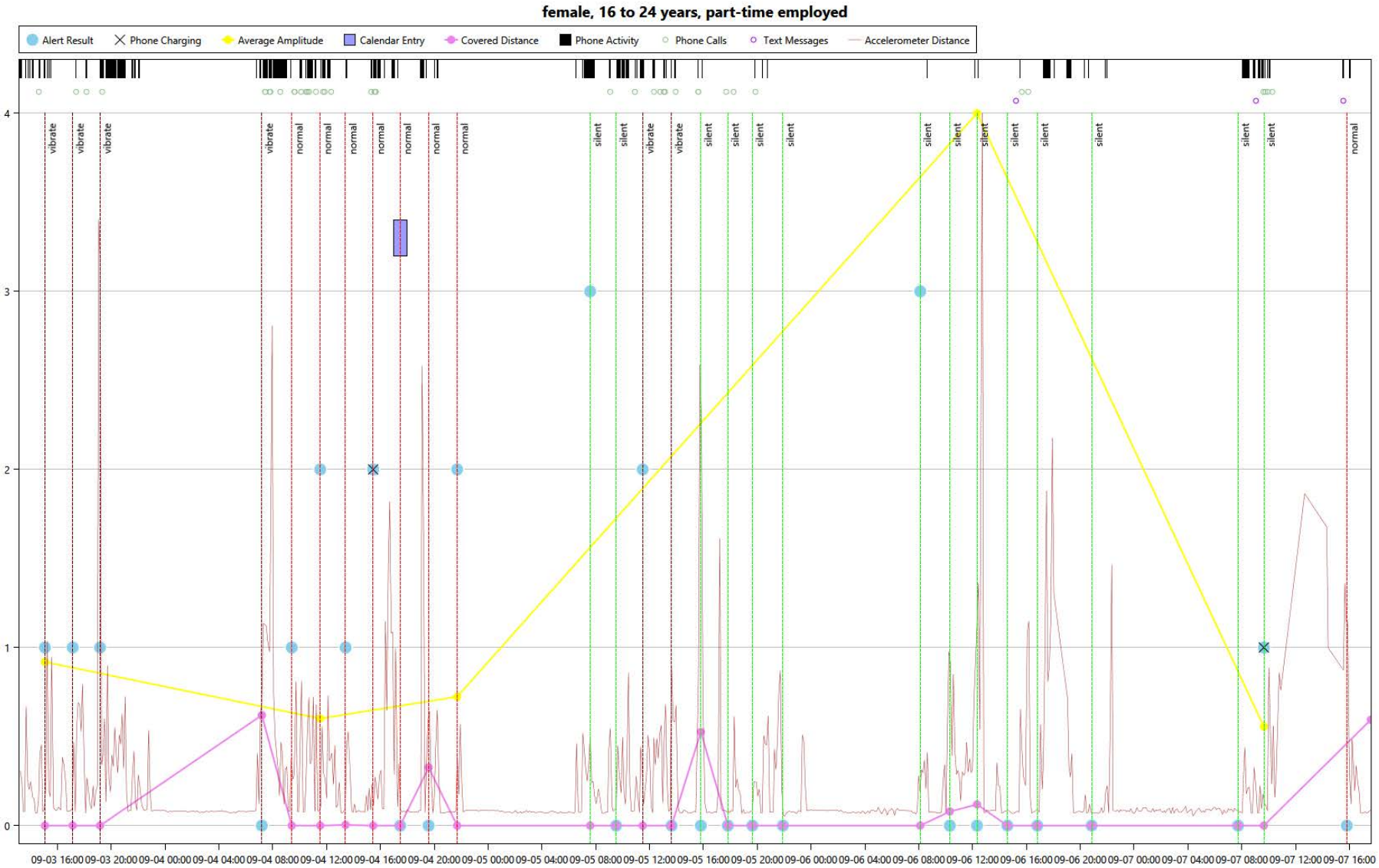

male, 16 to 24 years, in education
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
normal
4
3
2
1
0
08-07 20:00
08-08 00:00
08-08 04:00
08-08 08:00
08-08 12:00
08-08 16:00
08-08 20:00
08-09 00:00
08-09 04:00
08-09 08:00
08-09 12:00
08-09 16:00

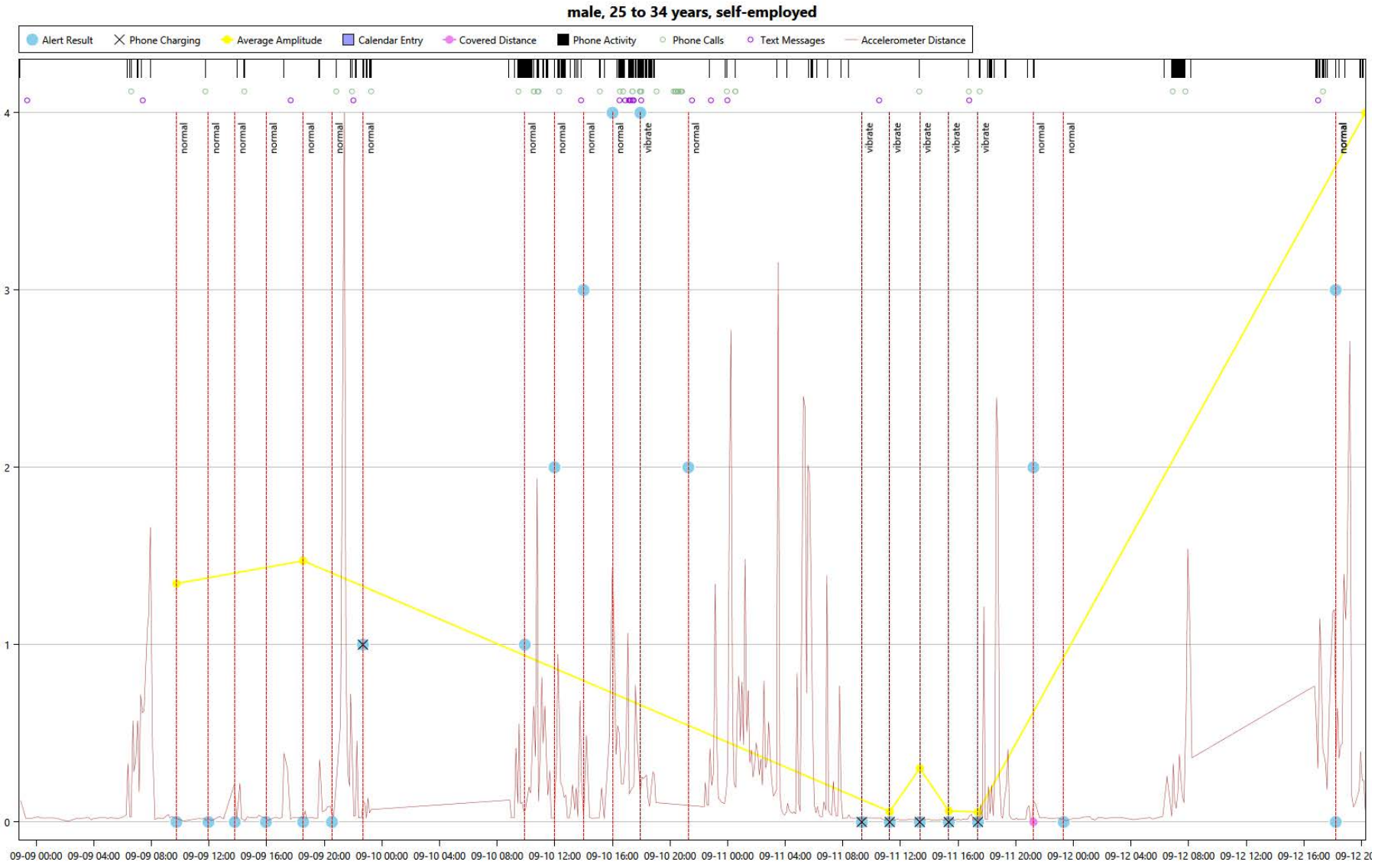

male, 25 to 34 years, part-time employed
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
normal
vibrate
0
1
2
3
4
09-08 00:00
09-08 08:00
09-08 16:00
09-09 00:00
09-09 08:00
09-09 16:00
09-10 00:00
09-10 08:00
09-10 16:00
09-11 00:00
09-11 08:00
09-11 16:00
09-12 00:00
09-12 08:00
09-12 16:00
09-13 00:00
09-13 08:00
09-13 16:00
09-14 00:00
09-14 08:00

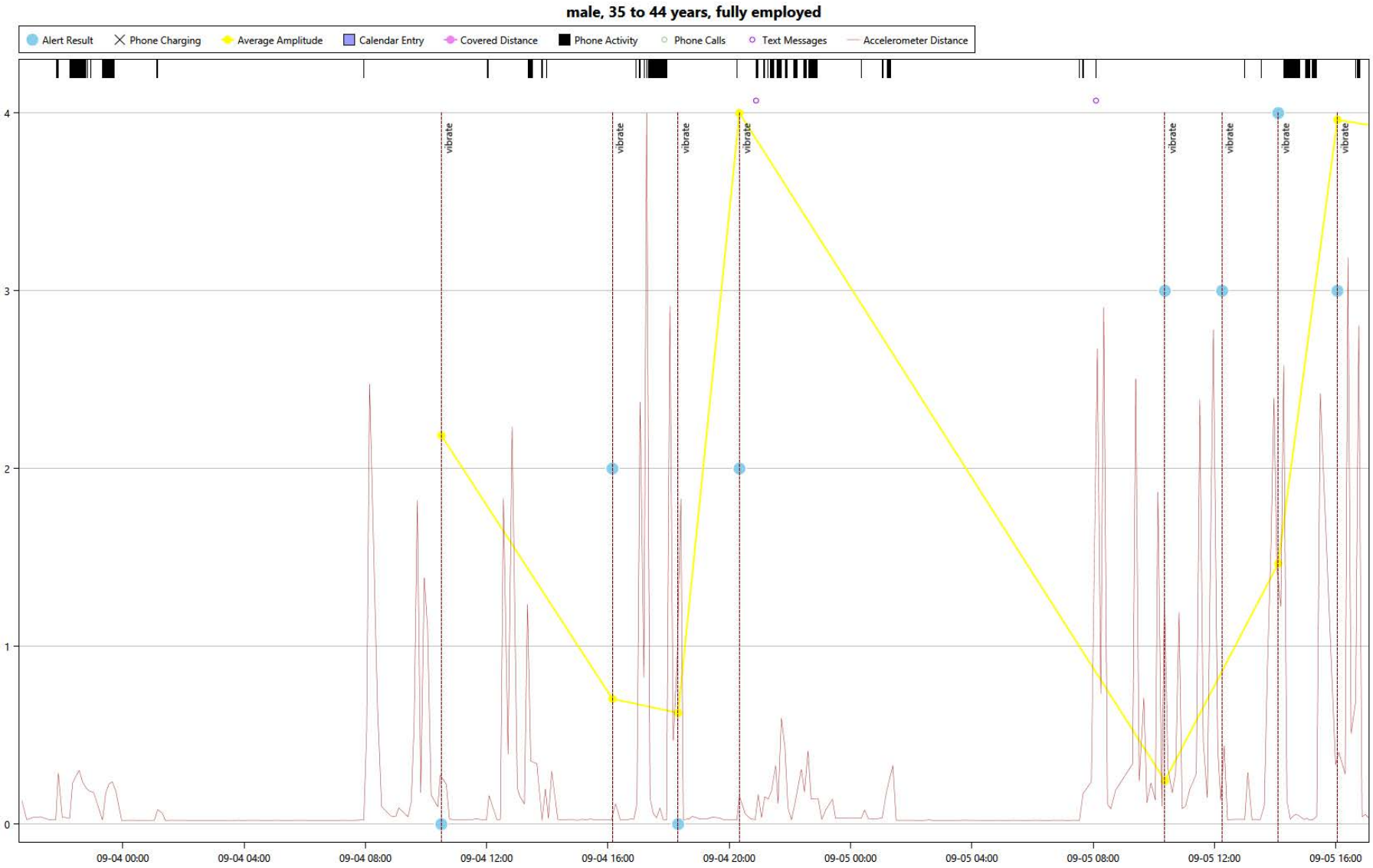

female, 25 to 34 years, in education
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
normal
4
3
2
1
0
09-05 16:00
09-06 00:00
09-06 08:00
09-06 16:00
09-07 00:00
09-07 08:00
09-07 16:00
09-08 00:00
09-08 08:00
09-08 16:00
09-09 00:00
09-09 08:00
09-09 16:00
09-10 00:00
09-10 08:00
09-10 16:00
09-11 00:00
09-11 08:00

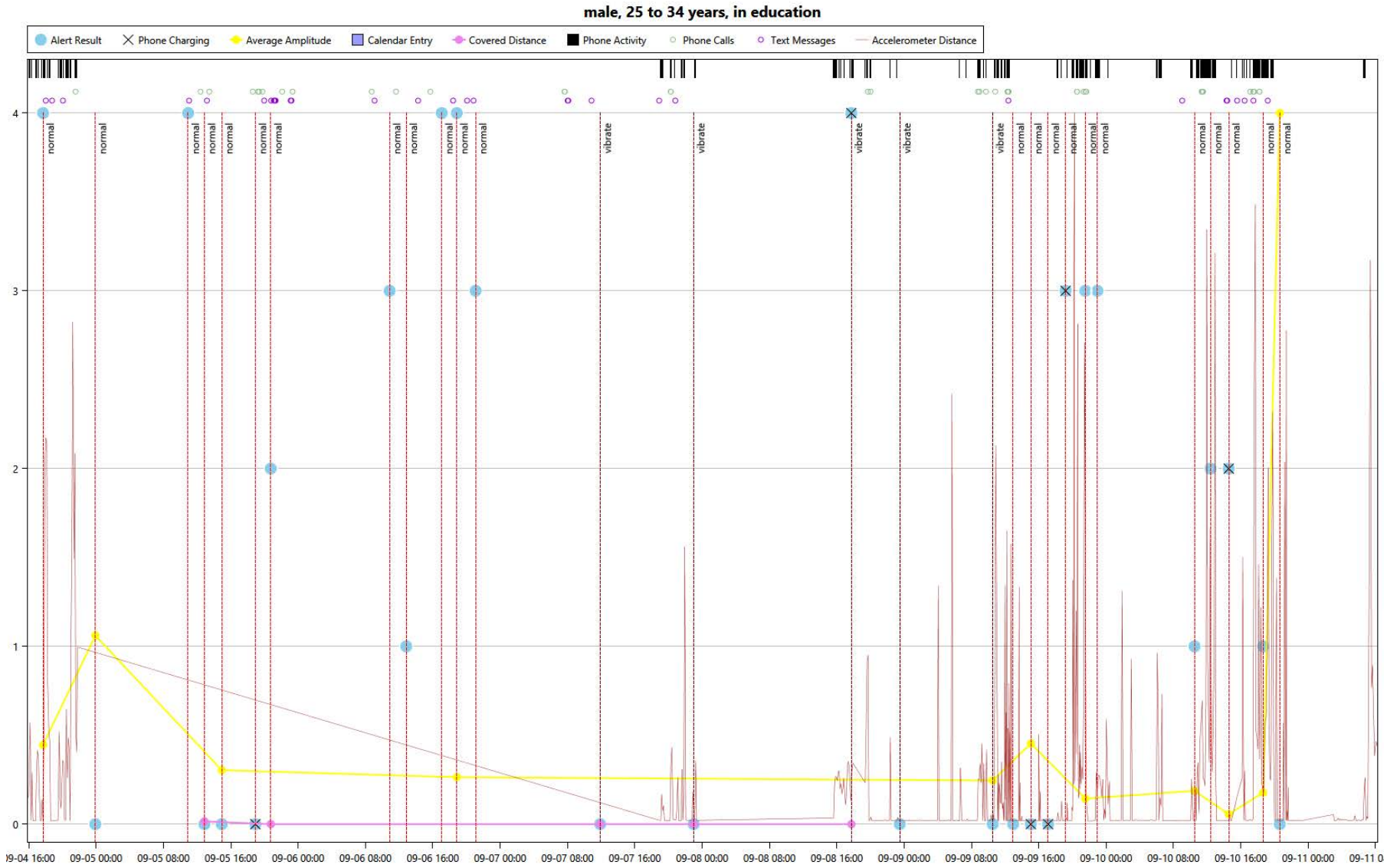

male, 16 to 24 years, in education
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
vibrate
4
3
2
1
0
09-04 00:00
09-04 08:00
09-04 16:00
09-05 00:00
09-05 08:00
09-05 16:00
09-06 00:00
09-06 08:00
09-06 16:00
09-07 00:00
09-07 08:00
09-07 16:00
09-08 00:00
09-08 08:00

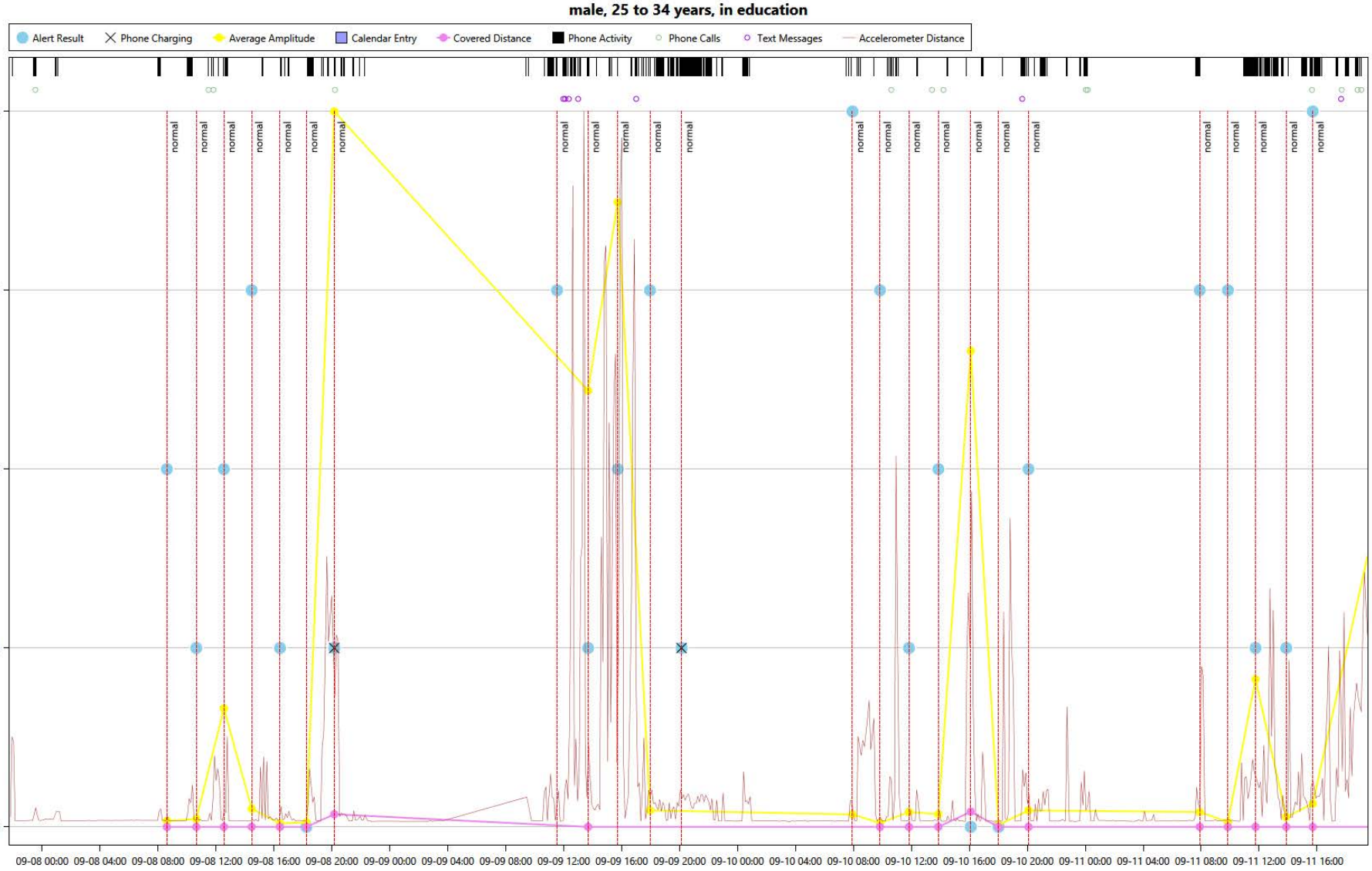

male, 25 to 34 years, fully employed
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
4
3
2
1
0
normal
vibrate
09-04 16:00
09-05 00:00
09-05 08:00
09-05 16:00
09-06 00:00
09-06 08:00
09-06 16:00
09-07 00:00
09-07 08:00
09-07 16:00
09-08 00:00
09-08 08:00
09-08 16:00
09-09 00:00
09-09 08:00

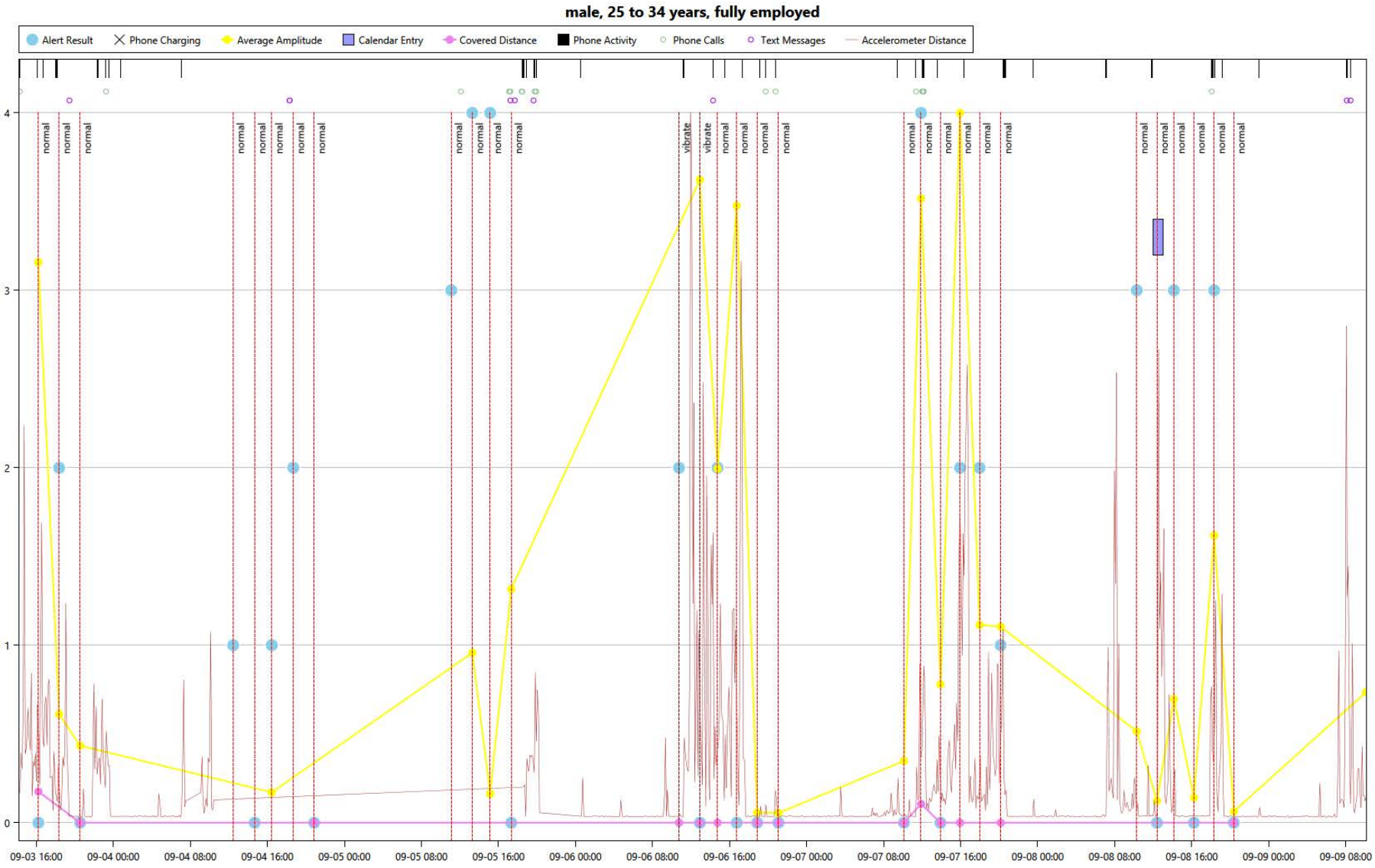

female, 16 to 24 years, part-time employed
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
4
3
2
1
0
vibrate
normal
silent
09-03 16:00
09-03 20:00
09-04 00:00
09-04 04:00
09-04 08:00
09-04 12:00
09-04 16:00
09-04 20:00
09-05 00:00
09-05 04:00
09-05 08:00
09-05 12:00
09-05 16:00
09-05 20:00
09-06 00:00
09-06 04:00
09-06 08:00
09-06 12:00
09-06 16:00
09-06 20:00
09-07 00:00
09-07 04:00
09-07 08:00
09-07 12:00
09-07 16:00

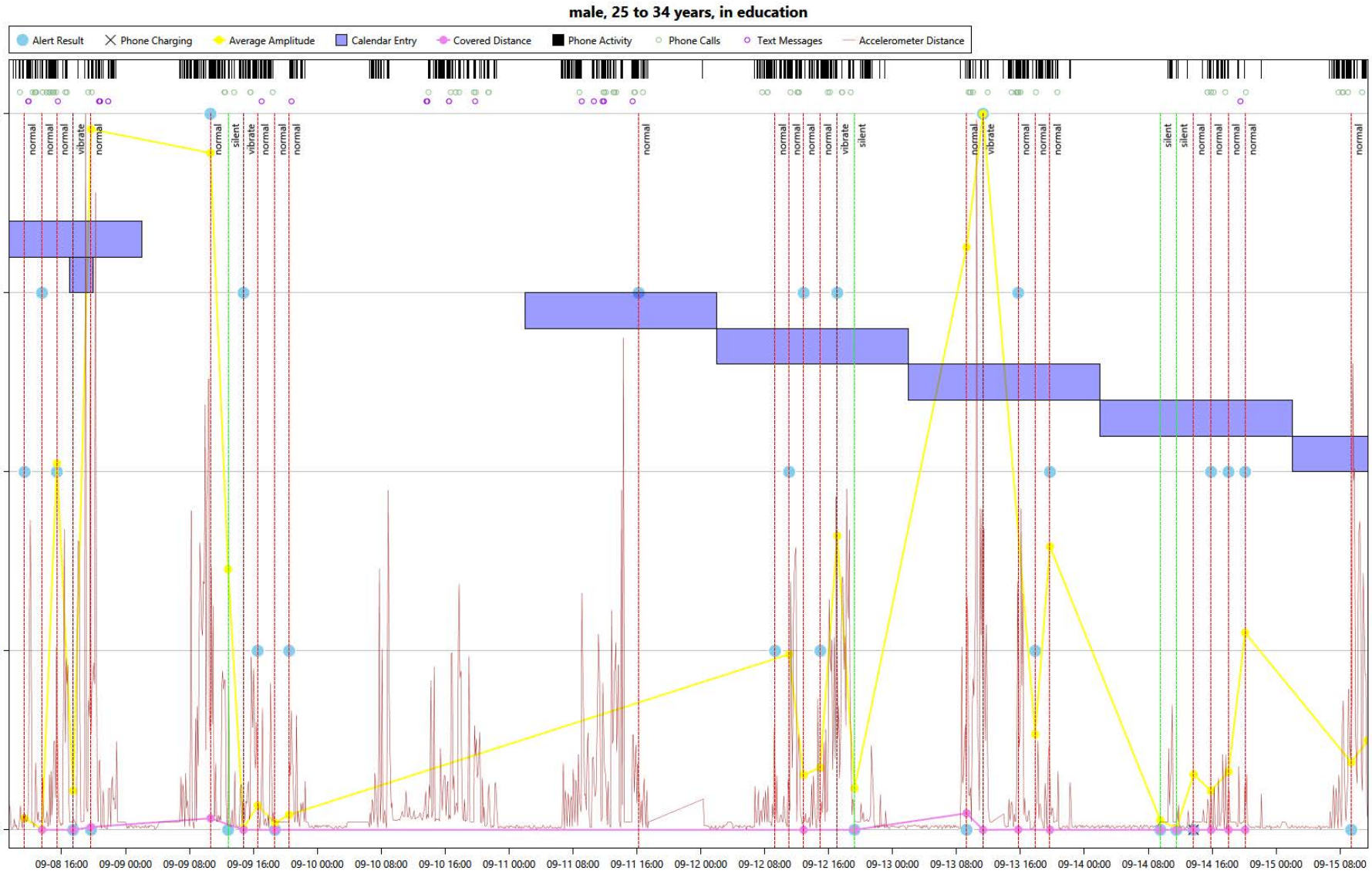

male, 25 to 34 years, self-employed
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
normal
vibrate
4
3
2
1
0
09-09 00:00
09-09 04:00
09-09 08:00
09-09 12:00
09-09 16:00
09-09 20:00
09-10 00:00
09-10 04:00
09-10 08:00
09-10 12:00
09-10 16:00
09-10 20:00
09-11 00:00
09-11 04:00
09-11 08:00
09-11 12:00
09-11 16:00
09-11 20:00
09-12 00:00
09-12 04:00
09-12 08:00
09-12 12:00
09-12 16:00

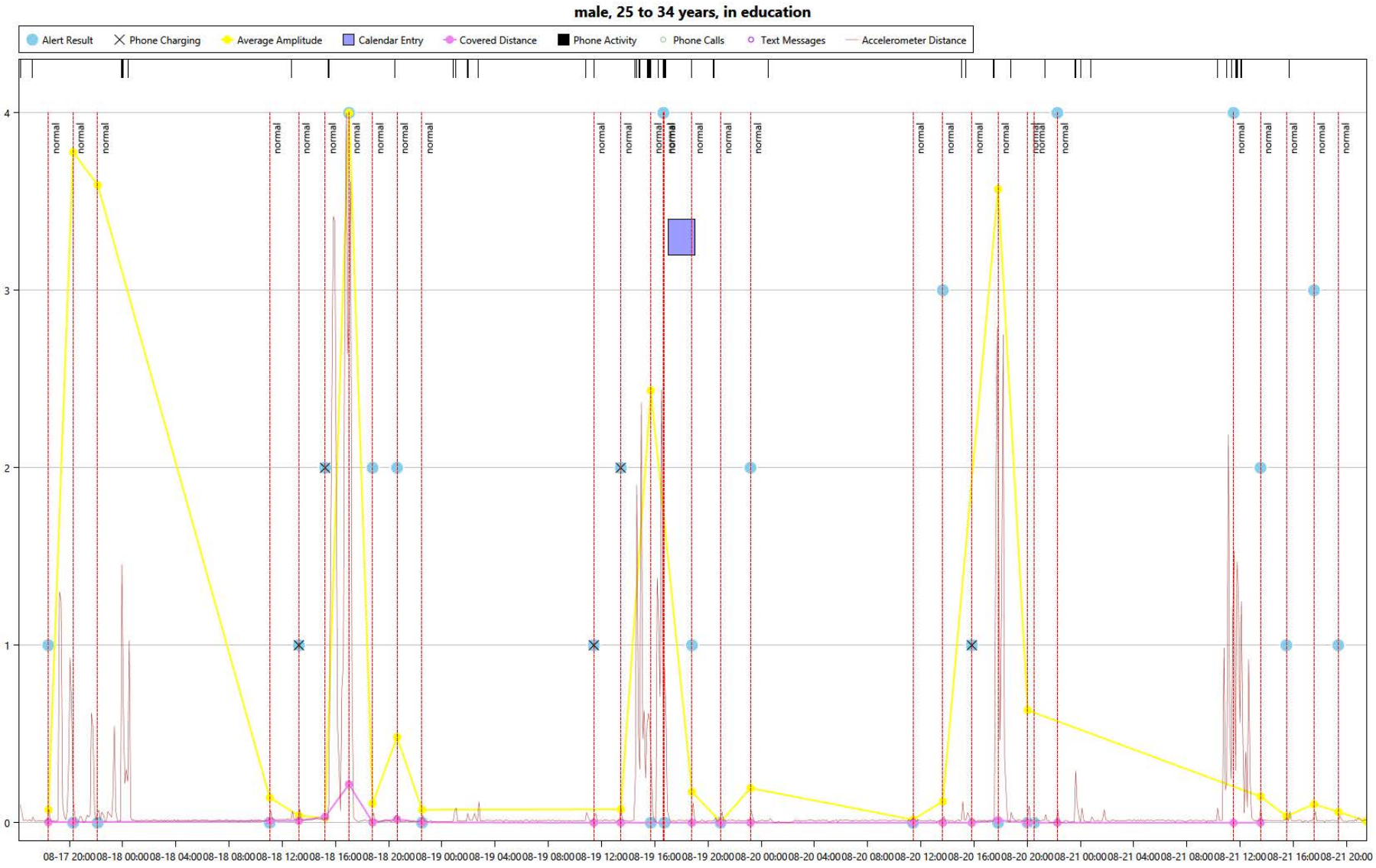

male, 35 to 44 years, fully employed
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
vibrate
4
3
2
1
0
09-04 00:00
09-04 04:00
09-04 08:00
09-04 12:00
09-04 16:00
09-04 20:00
09-05 00:00
09-05 04:00
09-05 08:00
09-05 12:00
09-05 16:00

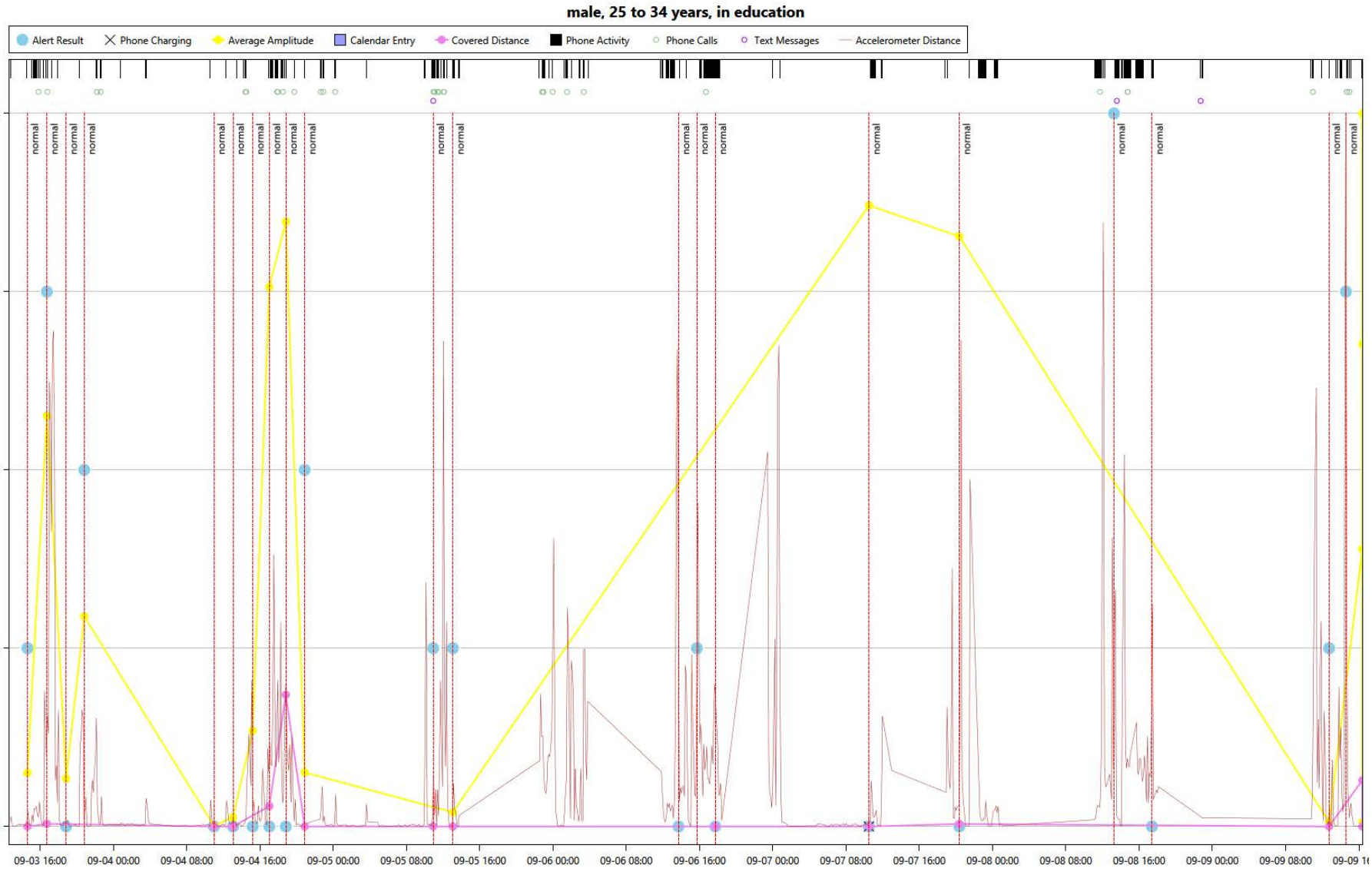

male, 25 to 34 years, in education
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
4
3
2
1
0
normal
vibrate
09-04 16:00
09-05 00:00
09-05 08:00
09-05 16:00
09-06 00:00
09-06 08:00
09-06 16:00
09-07 00:00
09-07 08:00
09-07 16:00
09-08 00:00
09-08 08:00
09-08 16:00
09-09 00:00
09-09 08:00
09-09 16:00
09-10 00:00
09-10 08:00
09-10 16:00
09-11 00:00

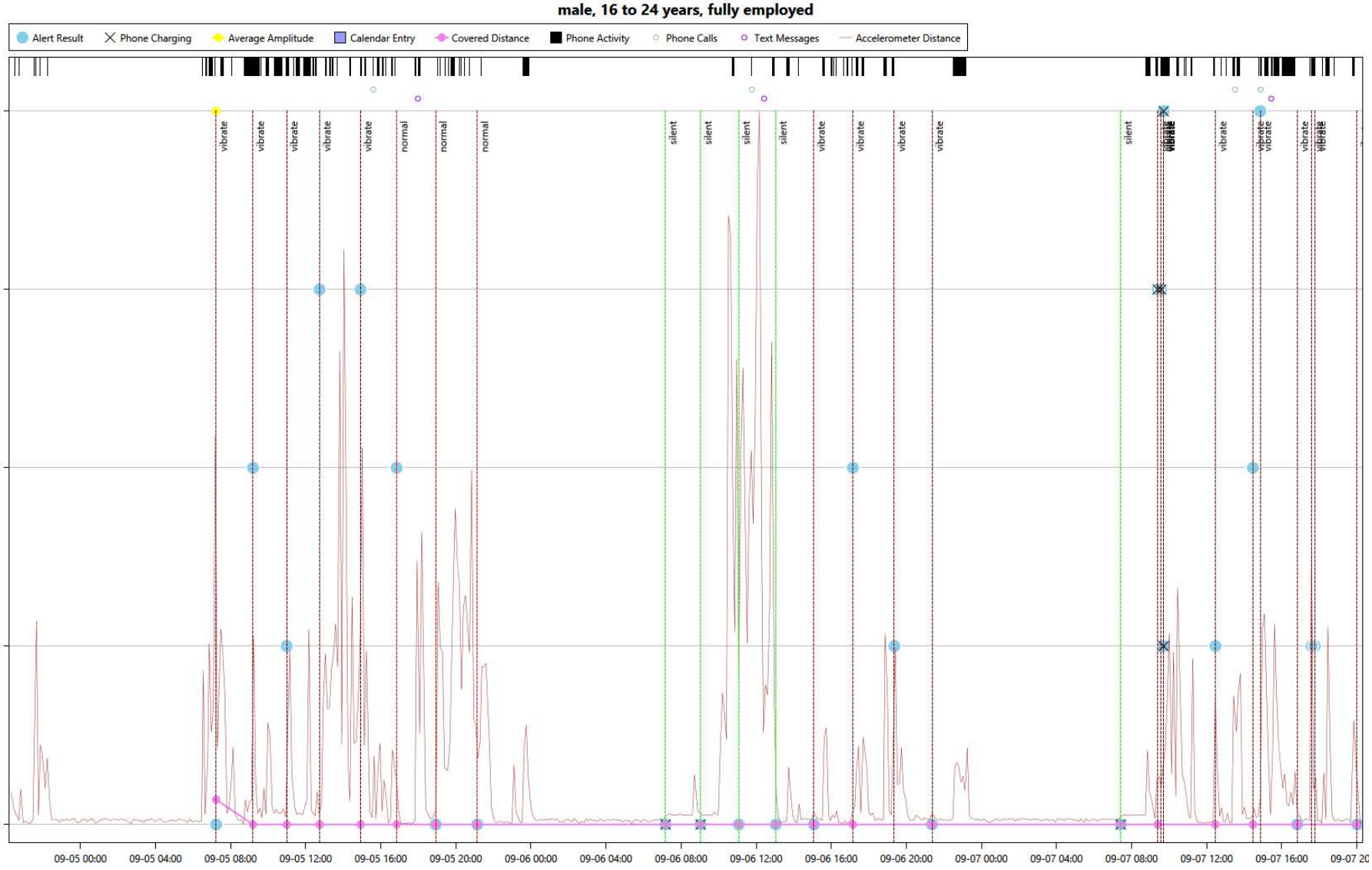

male, 25 to 34 years, in education
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
normal
4
3
2
1
0
09-08 00:00
09-08 04:00
09-08 08:00
09-08 12:00
09-08 16:00
09-08 20:00
09-09 00:00
09-09 04:00
09-09 08:00
09-09 12:00
09-09 16:00
09-09 20:00
09-10 00:00
09-10 04:00
09-10 08:00
09-10 12:00
09-10 16:00
09-10 20:00
09-11 00:00
09-11 04:00
09-11 08:00
09-11 12:00
09-11 16:00

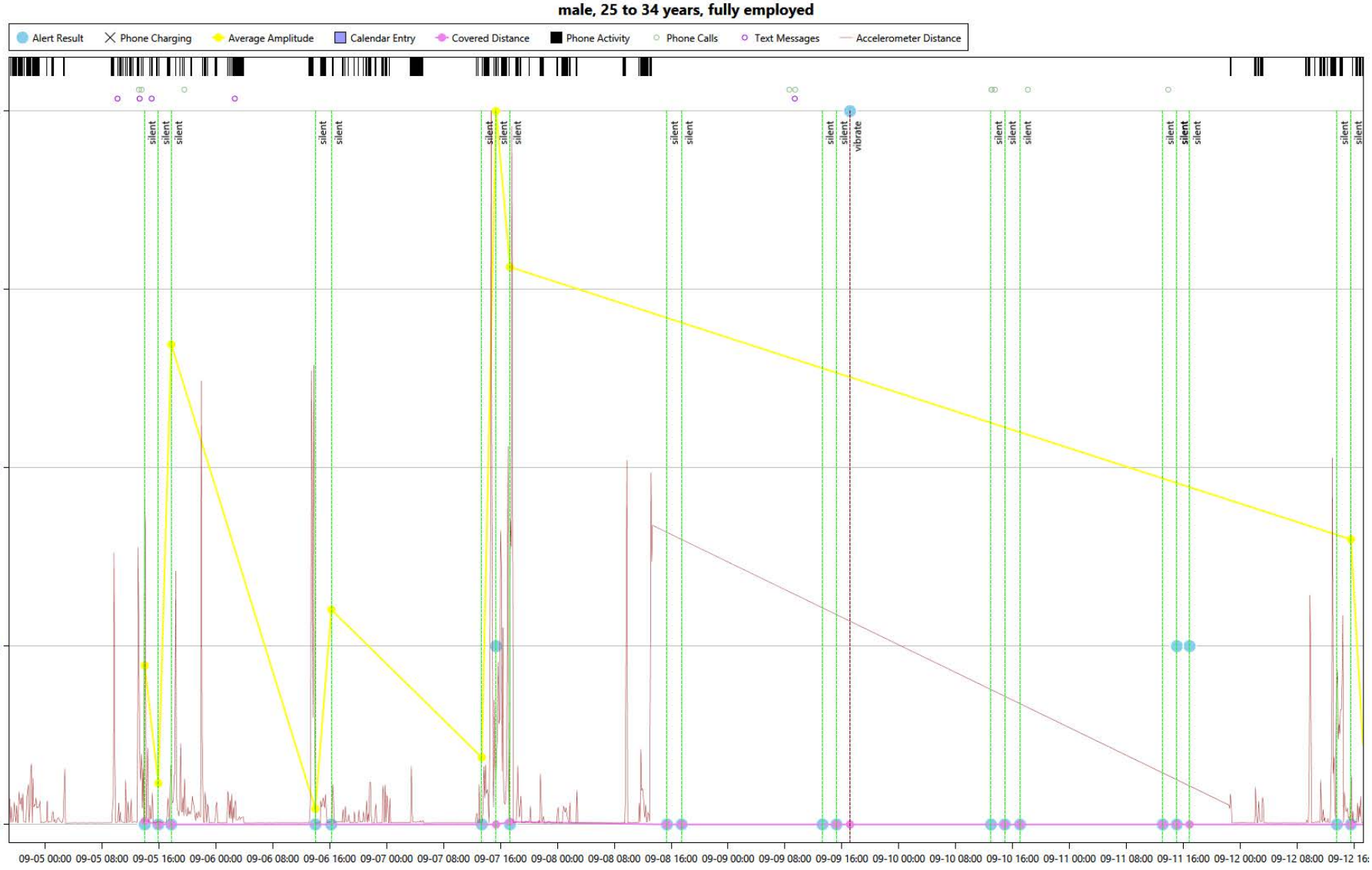

male, 25 to 34 years, fully employed
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
4
3
2
1
0
normal
vibrate
09-03 16:00
09-04 00:00
09-04 08:00
09-04 16:00
09-05 00:00
09-05 08:00
09-05 16:00
09-06 00:00
09-06 08:00
09-06 16:00
09-07 00:00
09-07 08:00
09-07 16:00
09-08 00:00
09-08 08:00
09-08 16:00
09-09 00:00
09-09 08:00

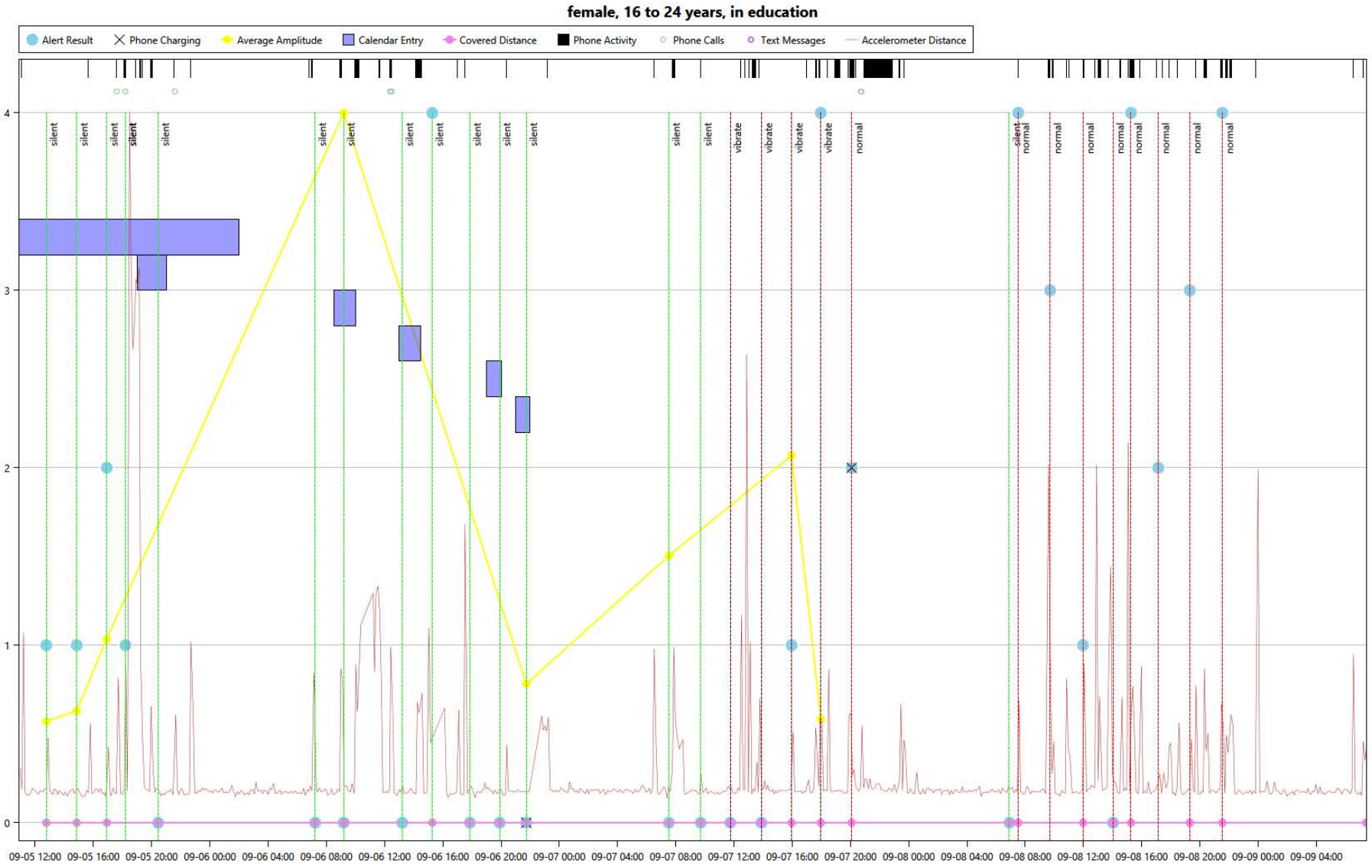

male, 25 to 34 years, in education
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
4
3
2
1
0
normal
normal
normal
vibrate
normal
normal
silent
vibrate
normal
normal
normal
normal
normal
normal
normal
vibrate
silent
normal
vibrate
normal
normal
normal
silent
silent
normal
normal
normal
normal
normal
09-08 16:00
09-09 00:00
09-09 08:00
09-09 16:00
09-10 00:00
09-10 08:00
09-10 16:00
09-11 00:00
09-11 08:00
09-11 16:00
09-12 00:00
09-12 08:00
09-12 16:00
09-13 00:00
09-13 08:00
09-13 16:00
09-14 00:00
09-14 08:00
09-14 16:00
09-15 00:00
09-15 08:00

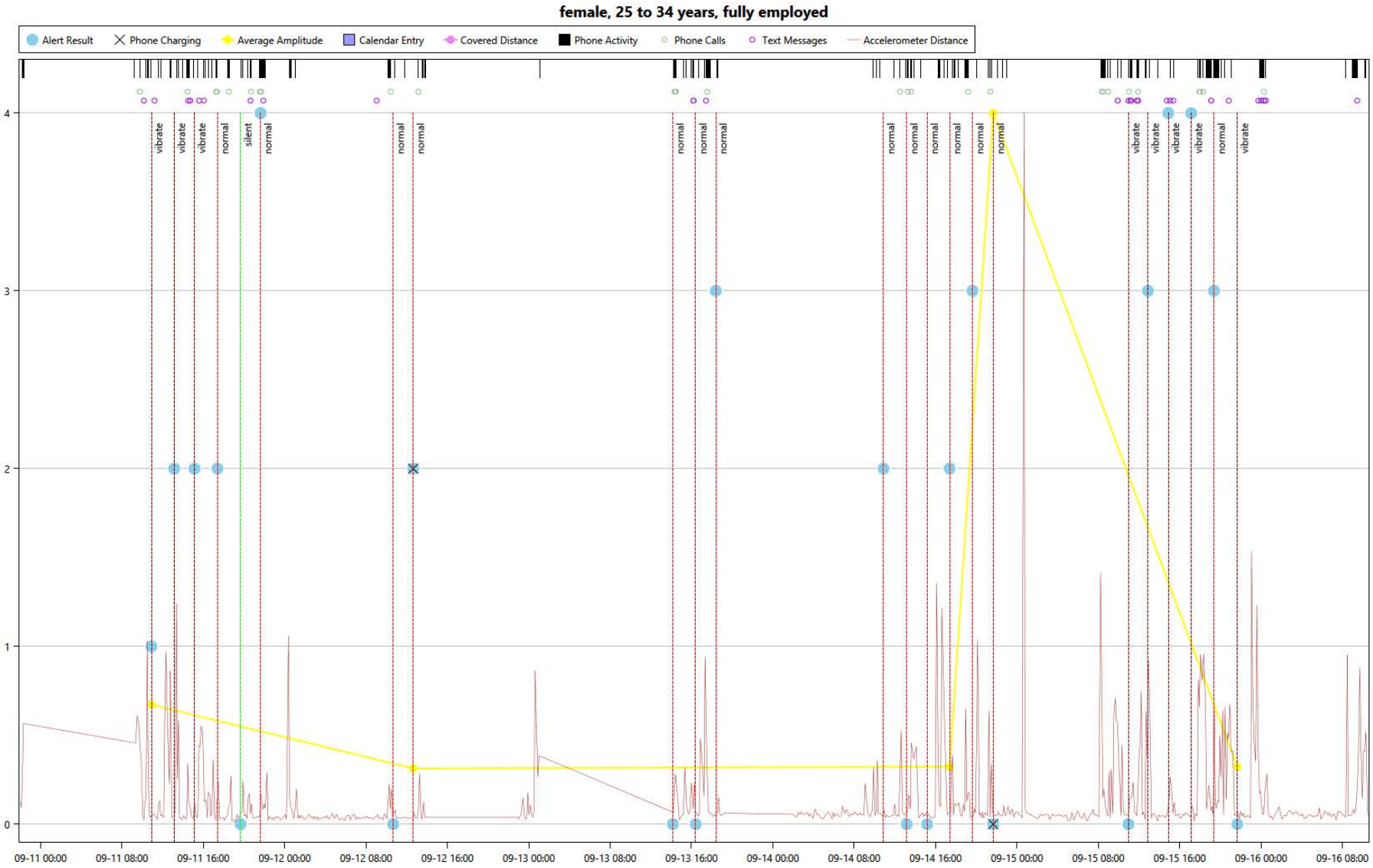

male, 25 to 34 years, in education
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
normal
0
1
2
3
4
08-17 20:00
08-18 00:00
08-18 04:00
08-18 08:00
08-18 12:00
08-18 16:00
08-18 20:00
08-19 00:00
08-19 04:00
08-19 08:00
08-19 12:00
08-19 16:00
08-19 20:00
08-20 00:00
08-20 04:00
08-20 08:00
08-20 12:00
08-20 16:00
08-20 20:00
08-21 00:00
08-21 04:00
08-21 08:00
08-21 12:00
08-21 16:00
08-21 20:00


male, 25 to 34 years, in education
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
normal
4
3
2
1
0
09-03 16:00
09-04 00:00
09-04 08:00
09-04 16:00
09-05 00:00
09-05 08:00
09-05 16:00
09-06 00:00
09-06 08:00
09-06 16:00
09-07 00:00
09-07 08:00
09-07 16:00
09-08 00:00
09-08 08:00
09-08 16:00
09-09 00:00
09-09 08:00


male, 16 to 24 years, fully employed
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
4
3
2
1
0
vibrate
normal
silent
09-05 00:00
09-05 04:00
09-05 08:00
09-05 12:00
09-05 16:00
09-05 20:00
09-06 00:00
09-06 04:00
09-06 08:00
09-06 12:00
09-06 16:00
09-06 20:00
09-07 00:00
09-07 04:00
09-07 08:00
09-07 12:00
09-07 16:00


male, 25 to 34 years, fully employed
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
silent
vibrate
4
3
2
1
0
09-05 00:00
09-05 08:00
09-05 16:00
09-06 00:00
09-06 08:00
09-06 16:00
09-07 00:00
09-07 08:00
09-07 16:00
09-08 00:00
09-08 08:00
09-08 16:00
09-09 00:00
09-09 08:00
09-09 16:00
09-10 00:00
09-10 08:00
09-10 16:00
09-11 00:00
09-11 08:00
09-11 16:00
09-12 00:00
09-12 08:00


female, 16 to 24 years, in education
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
4
3
2
1
0
silent
vibrate
normal
09-05 12:00
09-05 16:00
09-05 20:00
09-06 00:00
09-06 04:00
09-06 08:00
09-06 12:00
09-06 16:00
09-06 20:00
09-07 00:00
09-07 04:00
09-07 08:00
09-07 12:00
09-07 16:00
09-07 20:00
09-08 00:00
09-08 04:00
09-08 08:00
09-08 12:00
09-08 16:00
09-08 20:00
09-09 00:00
09-09 04:00


female, 25 to 34 years, fully employed
Alert Result
Phone Charging
Average Amplitude
Calendar Entry
Covered Distance
Phone Activity
Phone Calls
Text Messages
Accelerometer Distance
4
3
2
1
0
vibrate
vibrate
vibrate
normal
silent
normal
normal
normal
normal
normal
normal
normal
normal
normal
normal
normal
vibrate
vibrate
vibrate
vibrate
normal
vibrate
09-11 00:00
09-11 08:00
09-11 16:00
09-12 00:00
09-12 08:00
09-12 16:00
09-13 00:00
09-13 08:00
09-13 16:00
09-14 00:00
09-14 08:00
09-14 16:00
09-15 00:00
09-15 08:00
09-15 16:00
09-16 00:00
09-16 08:00