Int. J. Human-Computer Studies 79 (2015) 20–34

Contents lists available at ScienceDirect

Int. J. Human-Computer Studies

journal homepage: www.elsevier.com/locate/ijhcs

Interruptive notifications in support of task management ☆, ☆☆

Celeste Lyn Paul [a,*], Anita Komlodi [b], Wayne Lutters [b]

[a] Department of Defense, USA
[b] University of Maryland Baltimore County, USA

ARTICLE INFO

Article history:
Received 3 March 2014
Received in revised form
11 January 2015
Accepted 1 February 2015
Communicated by: E. Motta
Available online 7 February 2015

Keywords:
Interruption
Multitasking
Notification
Task management
User experience

ABSTRACT

Our research explores how interruptive notifications support task management in a desktop environment. We conducted two user studies with a community of open source software users and developers to explore their experience with interruptive notifications. We found that certain kinds of notifications support multitasking, task prioritization, task management, as well as influence task disruption management. We discuss how these behaviors affect the notification-task management user experience and offer design guidelines derived from these results to inform better design of systems that interrupt through notification.

1. Introduction

We live in an increasingly active information environment. As this becomes livelier and richer, the demands on our limited attention also increase. Continually checking for new information manually is tedious and time consuming. We require improved technology services that help us maintain awareness of updated information while mitigating the negative impact that interruptions can have by diverting our increasingly fractured attention. Interruptive notifications, such as alerts for the arrival of a new email, the completion of a remote backup, or a rapidly discharging laptop battery, are examples of notification services that help us maintain awareness of changing system state while allowing us to focus our attention on the other tasks at hand as they do not demand context switching, but rather communicate on the periphery. This research explores the complex environment that notifications exist in, and aims to characterize in more detail new and known factors that affect the interruptive notification user experience.

In our work, we define an *interruption* as the method of forcefully switching attention from one piece of information to another. *Notifications* are a type of information alert that informs the user of a system event or update. *Interruptive notifications* are notifications that intend to draw the user's attention in order to inform the user of a new event or information, such as a new chat message from a friend.

Notification displays are inherently interruptive because the system must divert the user's attention in order to deliver the information. These notifications usually appear in a small popup window in the middle or at the periphery of the screen. This type of notification is different from passive notifications that do not interrupt users in order to inform them, such as an email about a new comment on your blog waiting for you in the Inbox. Many application event managers and remote information services rely on interruptive notifications to deliver information updates to users in a timely manner.

Interruptions are a classic Human Computer Interaction (HCI) topic and there is continuing interest in understanding interruptions in multitasking environments in the HCI community (Gould et al., 2012). Although interruptive notifications can degrade user attention by presenting competing focal points, this does not always have a negative effect. More importantly, when done well, notifications can also support task management and decrease user anxiety about the system's current state. Notifications provide an important service that helps users manage tasks and changes in their information (Iqbal and Horvitz, 2010). The design of interruptive notifications presents a challenge because these services must deliver information to the user while balancing the costs of interrupting the user with the benefits of information awareness.

The goal of our research is to study how interruptive notifications support multitasking for a common class of users, *knowledge workers*, using their common work platform, a networked desktop computer. We chose to focus on knowledge workers specifically because their daily tasks are information rich and they routinely

☆Research was conducted while at the University of Maryland Baltimore County.
☆☆This paper has been recommended for acceptance by Henrik Christensen.
* Corresponding author. Tel.: +1 443 634 3755.
E-mail address: clpaul@tycho.ncsc.mil (C.L. Paul).

http://dx.doi.org/10.1016/j.ijhcs.2015.02.001
1071-5819/

multi-task between very different kinds of work behavior, from more solitary critical thinking to highly communicative collaboration with others. Knowledge workers are known to be particularly sensitive to increases of information and interruptions in their environment (González and Mark, 2004; Mark et al., 2005). Most work with integrated, general-purpose computing devices versus the task-specific tools used in other more production-oriented work domains. This allows us to study a host of interruptive notifications within the shared context of one operating system on one machine per user.

A key contribution of this paper is to offer a qualitative methodology to study the user's own notification environment. We present results from User Experience Reports (UXR) adapted to study interruptive notifications in a desktop environment. The UXR allows us to move out of the laboratory environment into the real world of a user's work while maintaining detail about an experience that is sometimes lost in more general field research methods. We conducted two field studies of open source software users and developers using the UXR. This research both validates and extends prior laboratory- and field-based findings about interruptive notifications, including the fact that task management is one of the most important uses of interruptive notifications in desktop-based knowledge work. The results describe the different ways users utilize interruptive notifications to support task management. Building on our understanding of this behavior, we then propose design guidelines to inform the design of more effective interruptive notification systems.

2. Related work

Interruption and notification is a wide research area that has been investigated in the domains of psychology, computer science, and interaction design. We summarize related work in these areas to provide background in what has been done to date and to frame the motivation for our own research. Specifically, we review work in multitasking and interruptions, the design of interruptive notification systems, and field-based methodologies used to study interruptions.

2.1. Multitasking and interruptions

There is a large body of HCI research focused on understanding the effects of interruptions and notifications on users' work processes. The focus of our research is on the role of interruptive notifications in task management, thus we review relevant notification and task management research.

2.1.1. Multitasking and interruptions

The presence of notifications implies that multiple tasks exist in the user's environment. The task a user is working on at the time of a notification is referred to as the *main task*, while the task initiated by the notification is the *interruption* task. A series of experiments by Gillie and Broadbent (1989) are foundational to our modern understanding of the effect of interruption on a task. Although their studies had conflicting results, they provide evidence that similarity and complexity of the interruption did have an effect on the main task.

Mark et al.'s (2005, 2008) work described similar conflicting results. In their earlier study (2005), the researchers examined the nature of fragmented work. They found that interruptions outside the user's work sphere, a collection of tasks related to a goal, were more disruptive than interruptions that were related to the work sphere. However, in the later study, Mark et al. (2008) directly compared the differences in interruptions related and not related to the user's work sphere and found no effects. Although, a related study by Ardissono et al. (2009) then found that interruptions related to the user's current work sphere *were* less disruptive than interruptions not related to the user's current work sphere. In our research we aimed to identify similar types of relationships that exist between the main and interruption tasks and to describe the impact of these relationships on user behavior.

2.1.2. Task prioritization

After receiving a notification, the user has to decide when and how to react to the notification. Work by Iqbal and Bailey (2008) found that users responded more quickly to interruption tasks if the interruption was scheduled as a breakpoint between main task chunks. This confirms previous work by Cutrell et al. (2001) that found users interrupted earlier in a task were more likely to request a reminder after being interrupted, as well as confirms work by Cades et al. (2007) regarding task complexity. Cades et al. found that the longer users work on a main task, the less recovery time is necessary when returning to the main task after attending to an interruption.

Content and saliency of the notification also have an effect on the user. Avrahami et al. (2008) found a number of factors that affected the amount of time it took for users to respond to an interruption, such as salience of the interruptive window and content of the interruption message. Users responded faster to interruptions that had more prominent interruption windows and longer interruption messages. This indicates that content of the interruption may have an effect on the disruption and perceived value of an interruption.

The value of an interruptive notification and the users' decision to when and how to respond to it was often determined by the context surrounding the notification event. In a field study that measured the effects of interruptions during various tasks, Vastenburg et al. (2008) found that interruptions with higher urgency were considered to be the most valuable type of interruption experienced by users. Their results identified urgency as the "primary indicator" for interruption acceptability. Context was the defining factor for determining interruption urgency, which translated into overall interruption value.

Another example of value and context is in a study by Paul et al. (2011a). This study found that participants reported a more positive user experience for interruptive notifications that provided information about social services than non-social services. Additionally, participants in this study seemed to value interruptive notifications more from certain social contacts over lesser important contacts. In our research we further examine the role of various notification characteristics and how they affect users' task prioritization.

2.1.3. Managing task disruptions

Notifications alert users of new tasks they may want to switch to and help them prioritize and structure their workflow. Sanders and Baron (1975) found that anticipation of an interruption may make users work harder to compensate for the cost of distraction the interruption creates. Giveska and Sibert (2004) replicated Sanders and Baron's results and described this phenomenon through the concept of compensation for interruption during a main task. Users who experienced more frequent interruptions adjusted their workflows by decreasing the amount of time away from the main task and by resuming the main task faster after interruption.

Iqbal and Horvitz (2007) provided additional insight as to why interruption compensation may occur. They found that users completed certain task interactions, such as paragraph completion while writing a document, more quickly immediately following an interruption than when performed with no interruption. They described this behavior as *task stabilization*, completing a task

cycle and preparing an interrupted task so it is easy to return to later. Andrews et al. (2009) also examined task stabilization in a study that tested pre-interruption alerts. In their study, they informed the user of an upcoming interruption by sending out a minor alert before the actual interruption. They found that users who received the pre-interruption alert were able to resume the main task faster than users who did not receive a pre-interruption alert.

Once users receive a notification and determine their next task action, they also often try to compensate for the interruption caused by the notification. Both Czerwinski et al. (2004) and Fogarty et al. (2005) informally observed task stabilization in their studies and found that users would first complete the current task interaction of the main task before attending to the interruption. Supporting task stabilization may be an important feature in an interruptive system. A smart system could help users complete small interactions that would normally be the source for main task errors, or to defer interruptions to between task interactions so that users do not need to stabilize tasks.

An experiment by Bogunovich and Salvucci (2011) explored the effects of time pressures on user-deferred interruptions. While participants tended to defer interruptions to a cognitive low point in their task, time pressures introduced additional interruption management strategies such as task stabilization to create a low point, and giving up and attending to the interruption during a cognitive high point if a potential low point is far away. In our research we examine strategies used by users to minimize the impact of an interruptive notification.

2.2. Designing interruptive notification systems

There are many ways to interrupt the user. A common way is through an interruptive notification message that draws their attention in order to deliver information. An interruptive notification system could be described in three conceptual components: information, behavior, and user interface. *Information* is the data that will be presented to the user by the notification system. *Behavior* is the logic and decision making executed by the system in order to determine how and in what way the user will be notified of the information. The *user interface* is the physical representation of the information according to the rules dictated by behavior.

2.2.1. Notification information

Information is an important aspect of a notification. Not all information has equal importance, urgency, or interest to the user. The context of the interruption can have an effect on the value of the information. However, even without fully understanding the complete context of the information and the interruption, general assumptions can be made about what information is important to users by developing models for use in design (Birnbaum et al., 1998).

Notifications should provide an adequate summary of information (Cadiz et al., 2002) so users can quickly determine how they should react to the notification. Although the information should be accurate and reliable (Berry, 2003), abstracting the system-provided data into a more compact and efficient message will make the notification easier to read “at a glance” (Matthews et al., 2004).

Notifications information should also be timely (Berry, 2003). Information delivered in a notification is often-time sensitive (Miller and Stasko, 2002) and delays in the notification information delivery could reduce the value of the information and increase the cost of the interruption to the user. For example, a notification informing the user of the current temperature is useless if the data is out of date or inaccurate, thus creating a low-value interruption to a user's main task.

Some research focuses on the design of certain types of information in notifications, such as incoming email, because of the frequent need to have timely updates on changes to this type of information. Information that is marked with high urgency or as newly arrived has been found to be an important type of information to users (van Dantzich et al., 2002). Importance is a contextual quality of notification information and can be difficult to define; but, specific sub-qualities of importance can be identified and used as a basis for assigning priority. For example, some of these related importance qualities include information marked with urgency (such as email), newly arrived information (such as an instant message), information that has met a time limit or expired (such as a task), or an expired bookmark (such as a file). Notification priority level (Matthews et al., 2004) can be used as a method for classifying information of varying levels of importance, allowing the notification system to decide how to best notify the user of new information.

2.2.2. Notification behavior

There are many guidelines available to help determine how the interruptive notification system interrupts the user and displays notification information. For example, van Dantzich et al. (2002) found that information such as sender and addressee of a message may be too much information in the notification user interface, and details such as those would be more valuable as hidden information available on demand. Gluck et al. (2007) described a method of dynamically highlighting important and relevant information in the notification as a way to help the user make sense of compact information. van Dantzich et al. (2002) also used color and symbols as a way to code and display more information in a notification. Cadiz et al. (2002) described the use of tooltips and mouse-over popups as a method for providing additional details.

This information design concept of progressive disclosure can be built on beyond simply providing additional information details to the user. Functionality such as links to other information, software, or related services in the system could be provided by the notification as well (Berry, 2003). Providing additional functionality helps the user react and respond faster to the information in the notification and possibly increase the value of the notified information.

The primary purposes of notifications are to provide awareness of changes in services outside the realm of the main task and to support multiple activities. Notification systems are meant to run as a support service while the user focuses on a main task (McCrickard et al., 2003). Sometimes the distraction of a notification acts as a trigger to switch main tasks (Iqbal and Horvitz, 2007). There are a number of ways disruption to the main task can be mitigated, and they suggest several ways for the user to more easily return to their main task. Saving the state of the main task could help the user resume the suspended main task more quickly and easily. Providing a reminder or indicator of the suspended main task could help the user resume the task sooner and mitigate distraction costs of the notification. Also, a playback feature could help remind the user of previous actions in the main task before the interruption.

Several researchers promote the ability for users to configure different aspects of the notification behavior and user interface in order to better tune the notification system to their needs (Cadiz et al., 2002; Berry, 2003). However, providing too much customization places an additional burden on users and forces them into the role of the designer (Nielsen, 1983, p.12). Instead, designers should analyze the errors and types of configuration options users can change to discover where the notification system is failing.

2.2.3. Notification user interface

The user interface of a notification has a big impact on the experience of the user interacting with the interruptive system. The most important notification information should attract the user's attention since the information could be of high urgency, time-sensitive, or great interest to the user. *Attentional draw* is defined as the "amount of attention attracted by an interruption's notification method" (Gluck et al., 2007). Attentional draw is often used to describe the features of a notification user interface that affect the user's attention. Motion is an animation method that has high attentional draw and can be used for delivering the most important and highest priority information (Matthews et al., 2004). Low priority notifications should have lower attentional draw than higher priority notifications. Static icons and other passive user interface elements are less disruptive methods that could be used for delivering less important and lower priority information (Matthews et al., 2004). Regardless of the priority level, the notification level must be easily recognizable in order for the user to quickly assess the priority of the information in the notification and respond accordingly (Berry, 2003).

It is important for information to be accurate and up to date and to reflect the current system status (Berry, 2003). Information that is frequently updated may be lower priority because frequent high priority notification would be distracting to the user. A consistent, but low attentional draw method is needed to provide the user with notifications of low priority information. Maglio and Campbell (2000) found that visual feedback is better than an auditory cue for indicating new information. The use of a notification indicator, "a passive device for conveying information status" (McCrickard et al., 2003) is a popular method for providing notification of status and updates to notification information as well as delivering notifications that require a low attentional draw (Cadiz et al., 2002; Matthews et al., 2004). For example, the notification of new software updates could be provided as an indicator in the task bar. Software updates are useful information to the user but not such a high priority that the user must be interrupted from the main task to attend to immediately.

2.3. Methods to study interruption

Field research is a common approach for studying interruptive notifications. Field methods allow researchers to study the notification experience in the user's natural environment. For example, many researchers have used observations and interviews with users in their natural settings to study interruption. The following two examples show how these methods can be beneficial in understanding users' natural notification-related behavior. Through observations and interviews with office workers, González and Mark (2004) studied task management in the workplace. They observed that work is very fragmented and that people manage tasks within a *work sphere*. These results have provided an enhanced understanding of the nature of interruptions and work processes. Iqbal and Horvitz (2010) collected metrics on how people used and responded to desktop notifications in an instrumented observational study. They collected information about participants' notifications on their own computers to understand notification and interruption behavior. Many of their participants used notifications as a way to maintain passive awareness of information updates and services. This knowledge allowed the researchers to think about new functionality to enhance desktop notifications.

Many researchers have also utilized experience sampling methods, a type of ethnographic method, to study notifications. Experience sampling methods are a way to collect and understand context, affect, and behavior in the user's naturalistic environment (Hektner et al., 2007). For example, Hudson et al. (2002) used an experience sampling method to record participants' reactions to random artificial interruptions. They found that participants found the interruptive notifications both useful and disruptive, depending on the context of the interruption and the participants' current tasks. Czerwinski et al. (2004) conducted a diary study to understand how interruptions affect task switching. Their results showed that there are a number of characteristics related to the task and the interruption that affect the difficulty of switching back to a task. The results were used to influence design in future information systems. Vastenburg et al. (2008) conducted a field-based experiment in which participants received generated interruptive notifications in their own home environment. Participants were asked to complete a questionnaire before and after a notification to evaluate the participant's notification experience. They found that urgency was the primary indicator for interruptive notification acceptance. More urgent notifications were much more likely to have high user acceptance than notifications that were not urgent. Our own early work utilized an experience report to study notification user experience (Paul et al., 2011a). A user experience report (UXR) is a specific type of experience sampling method aimed at capturing the user's experience. A UXR aims to capture details about an experience in the user's natural context soon after the event occurs. Many of the UXR questions were derived from the lessons learned of previously described research. We were able to find relationships between contextual factors, and used this experience to iterate and improve the study instrument for the method used in this paper's research.

3. Methodology

The purpose of our research was to study user behaviors surrounding the use of interruptive desktop notifications. Recall that we wanted to focus on a specific class of users, knowledge workers, who are particularly sensitive to interruptive notifications. Thus, we conducted two field studies that explored the notification experiences of users of and contributors to a large open source software community called KDE.[1]

In the first study we collected 235 experience reports in a User Experience Report (UXR) study. The UXR method was designed specifically to study context in the interruptive notification user experience. KDE users were recruited to gain an understanding of the KDE interruptive notification environment. We then followed the UXR study with a second study consisting of 14 semi-structured interviews that focused on one specific group of KDE users, software developers who contribute to the KDE project. The interviews allowed us to explore the major themes found from the UXR study in more depth. Participants in both studies were compensated with a small donation to the KDE non-profit organization on their behalf. The overall study design was such that the UXR and interview studies were complementary. Our goal was not to find overlapping results across two studies, but to use the interviews as second phase in our research in order to gain a depth of understanding in the most interesting themes identified in the UXR study.

3.1. KDE research setting

As noted earlier, much of the literature on the design and evaluation of notification systems has been conducted in a laboratory setting using simulated work tasks. Users in these studies are meant to be generic and representative, but all too often are Psychology and Computer Science undergraduate students at large research universities. As exemplified by Mark et al. (2005) there is great value in studying the user experience of notifications in the wild. It can also be

[1] www.kde.org

advantageous to focus on a particular community of users for whom their shared context is well known. We decided to take this approach by naturalistically capturing the actual work practices of members of a specific open source software community. The first author was a participant observer in the KDE open source software community, allowing her to deeply understand the culture, language, work demands, and software environments of its users. For the reader to best appreciate our KDE population and meaningfully interpret our results we provide additional background on this unique environment.

Open source is a software licensing philosophy that believes the human readable code source of software should be available for the public to freely install, modify, or redistribute. Open source can also refer to the community and development practices of thousands of free/libre/open source software (FLOSSPOLS, 2006) projects that subscribe to this philosophy and license their software under one of the many available software licenses. Open source projects tend to have communities that support highly distributed development and rely on online communication tools to support collaboration and development. People contribute to open source projects for many reasons, including as a place to gain experiences, an outlet for technical creativity, a place to join a community of like-minded people, and for material or monetary remuneration (Lakhani and Wolf, 2005).

KDE is a very large open source software community of contributors (developers, designers, translators, etc.) and users that produce a desktop environment and software suite similar to Microsoft Windows and Apple Mac OS X. KDE contains over 6 million lines of code written by more than 1800 contributors through 1 million code changes.[2] KDE contributors work together in a globally distributed virtual environment to produce code, maintain software, and promote the project. KDE is used all over the world by a wide variety of individual users, governments, and schools.

KDE has its own notification system through which all KDE applications send messages. This system forces all notifications to have the same look and feel, constrains notification behavior to the same set of rules, and manages notifications through a common configuration system. It may be useful to note that this integrated notification system in some ways more closely resembles the conceptual designs of time share minicomputers than our current desktops which had historically branched off to allow individual applications the ability to control their own notification user interfaces. Currently there is a move away from these messy, inconsistent heterogeneous systems back toward unified designs like KDE. This can be noticed in the operating systems that run most of our portable devices such as Android, iOS, and the active tiles in Windows Phone.

We chose to study KDE notifications and the KDE community for several reasons. First, KDE has an integrated notification system that ensures a certain level of user experience consistency in the KDE environment. All notifications, regardless of the application or service that sends them, have a consistent style for writing and presenting information, behavior, and user interface look-and-feel. Additionally, all participants have similar notifications experiences even if they report different experience contexts.

Specific details of the KDE notification system include:

- All applications and services send notifications through the same system that then displays a notification to the user. This results in a consistent look and feel across all notifications.
- Notification user interface supports the addition of action buttons that allow the user to click and take an action (such as begin software updates or delete an email) without switching away from the current application to the application or service that sent the notification.
- An icon indicator on the task bar provides a visual indicator when users miss a notification and acts as a link to notification history and system configuration.
- A history of recent notifications is available by clicking the notification indicator in the task bar.
- The state and progress of a task linked to the notification system is available by clicking the notification indicator in the task bar.
- The KDE notification system is highly configurable; users have the ability to customize the conditions to receive a notification and to temporarily turn notifications on/off.

Second, KDE users and contributors are more technically competent than average users. This technical competency may provide them the language and understanding to better describe their experiences than less technical users. However, there is the drawback that these types of users may focus too much on details about the implementation and technology rather than the experience. Third, software developers are a modern type of knowledge worker who often work with others in a local or virtual space. This new work context results in additional notifications from supporting virtualization technology such as cloud-based development tools and social communication services. This virtualized environment makes them an ideal audience for study because they would benefit greatly from an improved interruptive notification system. Finally, many members of the KDE community were interested in participating in our research because the results would be a valuable contribution to the project. In the short term, the donation made on their behalf for participating in the studies would help fund organization activities and services that they care about. In the long term, results of this research would provide user-centered feedback and recommendations on how to improve the notification system.

One drawback to engaging with the open source community is its lack of gender diversity. While the number of women in science, technology, engineering, and mathematics careers is approximately 26% (NSF, 2011), the number of women who contribute to open source projects is estimated at 1.5% (FLOSSPOLS, 2006). The number of women contributors in the KDE community is estimated at 10% (KDE 2012). Although the number of women in the KDE community is above the average FLOSS project, it is well below the normal and STEM population distributions. Therefore there are limitations in how far this work can generalize gender similarities or differences.

3.2. User experience reports

Modern design practices have shifted away from strictly a *usability* approach-efficiency, effectiveness, and satisfaction (ISO 9241-11:1998) measured in a laboratory setting-to a more holistic consideration for the entire user experience. *User experience* is defined as a user's "perceptions and responses that result from the use of or anticipated use of a product, system, or service" (ISO 9241-210:2009). Good user experience is a critical factor in user performance, user enjoyment, user acceptance and continued use of a system (Beaudry and Pinsonneault, 2010; Forlizzi and Battarbee, 2004). However, understanding user experience is highly dependent on understanding context. Factors within that context can be critical to successful system design, such as an interruptive desktop notification system.

Field methods are well-suited for studying context, although each have their strengths and weaknesses when studying interruptive notification environments. We decided to utilize the User Experience Report (Korhonen et al., 2010b) to mitigate some of the methodological weaknesses of these methods. A User Experience Report (UXR) is a type of user experience sampling method that

[2] KDE Development (Wikipedia) http://en.wikipedia.org/wiki/KDEDevelopment.

utilizes a semi-structured report completed in situ soon after the participant experiences the study phenomena (Consolvo and Walker, 2003; Korhonen et al., 2010a). The participant completes a survey immediately after a specified experience with an interactive system, in our case an interruptive notification.

The UXR has three main benefits to interruptive notification research:

- **Context:** The instrument contains structured questions to record the user experience and the data collected is provided by a user rather than 3rd party observations or logging.
- **Naturalistic:** A user's own notifications in their own environment are sampled rather than artificially generated notifications in a laboratory environment.
- **Timely:** Experiences are captured at/near the time that they occur rather than a reflective report by the user of a past experience.

We chose to adopt and modify the UXR as a new interruption experience sampling method rather than utilize one of the methods utilized in related interruption and notification research because no single method seemed well-suited to study the complex context of this user experience. The use of interactive systems tends to be a collection of many experiences that culminate into a large overall user experience of the system. The UXR offers a way of studying these smaller experiences to study their effects on the greater system.

A UXR is similar to a diary study in that it is a sample experience from the user's natural environment. However, there are several key differences. For example, a UXR is different from a diary study in that it is not an ongoing record of experience. A UXR only samples a single experience from a single participant where a diary study is a longitudinal sampling of many related experiences from a single participant. Diaries contain reports of multiple overlapping experiences while the UXR is anchored on a single experience. Sampling a single experience makes an in-depth description of the experience possible. A participant would experience study fatigue if they had to provide a large amount of detail about experiences over a long period of time.

Participating in a UXR study requires little effort on the part of the participant even though the responses are more involved than a single diary study entry. Diaries are kept through the day or summarized at the end of the day over a period of many days. The participant completes the UXR survey immediately after the experience and each participant only reports one experience. For example, a participant may be notified of a new email, read the new email, complete the UXR, and then return to her task. The burden of participation is greater in a diary study that requires repeated, regular participation over a period of time. It is not uncommon for the number and quality of responses in a diary study to degrade over the course of the study (Paul et al., 2011b).

A UXR is more structured to support user feedback and collect targeted responses where a diary study usually requires less structure to support open-ended responses. Also, a UXR is executed so that it is completed soon after an experience, leading to more reportive responses. The completion of a diary entry is often scheduled by the user and can result in more reflective responses. When insights over a period of time are not critical to the research, a UXR offers more benefits than a diary study.

3.2.1. Study instrument

The UXR contains a mix of qualitative open-text responses as well as quantitative binary (yes, no), orthogonal (e.g., male, female), and ordinal (e.g., age range) response and implemented in a web-based survey. The survey form also administered the informed consent and instructions. The UXR included closed and open-text questions related to the notification, the task that was interrupted, and about themselves including their work environment. Participants were asked to provide the following information in the UXR (Fig. 1):

3.2.2. Participants

As noted earlier, the study of KDE users is one of the novel factors of our research. We collected 235 UXRs from participants who were recruited through KDE community social media such as Twitter, mailing lists, and IRC. Participants were compensated with a €1 donation to the KDE non-profit organization for each completed UXR.

There were more male participants than female participants; however, this ratio is normal for open source software populations (Male=215, Female=7, No Answer=13). Most participants were in the 18–24 (n=64) and 25–34 (n=112) age bracket (35–44=31, 45–54=13, 55+ =3, No answer=12). Most participants had a College or Graduate-level education reflecting a highly educated population (High School=38, Some College=34, College=79, Graduate=71). Most participants were in the technology field, either through education or employment (e.g., Software Developer=65, Student=59, Academic/Researcher=15).

The KDE community is a very international project with contributors on nearly every continent. The study population reflects this with

Notification User Experience Report

About the Notification

1. Upload a screenshot of your KDE desktop as it was when you received the notification.
2. What was the notification about? (open text)
3. What service or application sent the notification? (open text)
4. Would you want a notification like this again in the future?
   a. Yes, No, It depends (selection)
   b. Why or why not? (open text)
5. Using one word, how would you describe your overall notification experience? (open text)

About your Task

6. What were you doing at the time of the notification? (open text)
7. Did you stop what you were doing when you received the notification?
   a. Yes, No (selection)
   b. Why or why not? (open text)
8. Was the notification related to the task it interrupted?
   a. Yes, No (selection)
   b. Please explain. (open text)
9. What best describes your use of the computer at the time of the notification:
   a. At work, Working from home, At school, Doing homework, At home, Other (selection)
   b. Please explain Other. (open text)

About Yourself

10. Job Role/Title (open text)
11. Age (selection)
    a. 18-24, 25-34, 35-44, 45-54, 55-64, 65+
12. Gender (selection)
    a. Male, Female
13. Country (open text)
14. Computer (selection)
    a. Laptop, Desktop, Tablet, Phone
15. English Language Proficiency (selection)
    a. Native speaker, Fluent (business-level), Conversational (tourist-level), Little/none
16. Education Level (selection)
    a. High school
    b. College (AA, BA, BS)
    c. Graduate (MFA, MS, PhD, MD, JD)
17. Role in KDE (multi-select)
    a. Developer, Contributor, Supporter*, User

* Supporter is a membership type in the KDE community

Fig. 1. User Experience Report used in KDE notification experience study.

similar levels of international participation, primarily in Europe and the Americas. Participants were globally distributed with then most from Germany ($n=46$), Spain ($n=23$), and the United States ($n=23$). Most participants were German, which makes sense since KDE is a German-founded project. The remaining participants were from other countries in Europe, North and South America, Africa, India, China, and Australia. (See Fig. 2).

Most participants were fluent in English (Tourist-Level=66, Fluent=111, Native Speaker=37, No Answer=10). Those who marked Little English ($n=11$) language skills still had well-written, comprehensible responses.

There are many roles within the KDE community. Some people are Developers ($n=19$) while other people contribute in other ways (such as translations or user interface design) ($n=34$). Users ($n=209$) can sometimes become Supporters ($n=38$) of KDE by providing a monetary donation or actively promoting and evangelizing the project. Sometimes participants were more than one, such as a Developer and a Supporter. Most participants in the UXR study were Users (inclusive of Developers, Contributors, and Supporters).

Participants were asked what type of computer they were using at the time of the notification reported in the UXR. The number of participants who used Desktops ($n=109$) and Laptops ($n=116$) was nearly equal (No Answer, $n=10$). Most participants were at Home ($n=128$) or Working from Home ($n=47$). Participants responded from Work ($n=35$), School ($n=3$), or some Other location ($n=12$) (No Answer, $n=10$).

3.2.3. Procedure

Participants were asked to complete a structured report shortly after a notification experience. The report questions were designed to collect information that would help us understand the context of their notification user experiences in four characteristics: user, user's environment, user's tasks, and interruptive notification. These characteristics were used as a framework to help understand the interruptive notification user experience.

Submitting a desktop screenshot was optional. Participants were instructed to minimize or hide personal information that was visible in their screenshots. The study instructions suggested minimizing windows with personal information; some participants used graphics programs to censor personal information with color blocks or blurring. This procedure was not unfamiliar for many participants since sharing screenshots of personal desktops is a common activity in open source communities. Self-censoring may have led to changes in the screenshots that could have affected the analysis and interpretation of the screenshot data. This was an acknowledged and accepted limitation to obtain the richness of personal desktop screenshots while preserving the safety and privacy of participants.

3.3. User interviews

Interviews with 14 KDE software developers were conducted as a follow-up study to the UXR study. KDE community software developers were targeted in the interviews as a way of exploring knowledge working in depth. Participants were recruited from known contacts within the community and in IRC channels that KDE developers frequented. Several participants were referred to by personal contacts and other interview participants. Participants were compensated with a €5 donation to the KDE non-profit organization.

The interviews were conducted as a way to follow up on major themes that emerged from the UXR analysis as well as an opportunity to explore the notification user experience in more depth from the perspectives of software developer-type knowledge workers. Participants were asked about their experiences with notifications in social contexts, use in task management, and general user experience.

Interviews were conducted over IRC chat service and lasted between 45 and 90 min. IRC was an appropriate communication mechanism for this community because many participants spoke English as a second language and had better English reading/writing skills than speaking skills. IRC is also the primary communication and collaboration mechanism for the KDE community and participants were very comfortable communicating using this medium.

Participants were all male ($n=14$) who were mostly in the 25–34 age bracket (18–24=1, 25–34=9, 35–44=3, 45–54=1). Most participants had a college or graduate degree (High School=1, Some College=3, College=6, Graduate=4). Most participants worked in technology, such as a Software developer/engineer ($n=5$), Student of computer science ($n=4$), and a Community manager for a software project, Chief Technology Officer, worked in information technology, an office worker, or a sales manager ($n=1$, respectively). Most of the participants were KDE Contributors ($n=12$) with many acting as developers for KDE ($n=7$). If participants were not KDE developers, then they were software developers in their profession. If participants were not software engineers for their profession, then they were

Fig. 2. Map of participant locations represented in the User Experience Report study.

Developers in the KDE community. Participants were from countries in Europe (two each from Denmark, Switzerland, and the United Kingdom; one each from Finland, Germany, Netherlands, Romania, Scotland, and Spain) and United States ($n=2$).

3.4. Analysis

Iterative inductive coding (Patton, 2001) was conducted on the UXR open-text responses in order to identify and develop major themes within the data. Categorical coding was used to develop details within the major themes. The logs from the user interviews were analyzed using a structured coding approach using the major themes from the UXR study. The interview data was analyzed in this way for the purpose of confirming or contradicting findings from the UXR.

Coding for the data was iterative with several passes to ensure thorough coding. Two researchers each coded 100% of the UXR open-text responses and user interviews. They met to review differences in coding structure, discussed common and differing themes, and merged common ideas into a single code when appropriate. The iterative and collaborative nature of theme resolution meant that we could not accurately calculate inter-coder reliability as it is traditionally done.

4. Results

The findings reported in this paper are the result of development of themes related to task management behaviors in an interruptive notification environment. To provide context of interruptive notification environment and the users that were studied, we first provide a summary of relevant UXR responses. We then discuss in depth the major themes related to task management based on our analysis of the combined analysis of the UXR and interviews.

4.1. Summary of UXR responses

The UXR instrument consisted of questions that required open-text and selection-based responses. This section summarizes results from the selection-based responses to provide context of the study environment and participants.

4.1.1. Tasks and notification environment

Participants received notifications from a variety of applications and services (Q3, coded). The most common sources of notifications were from social media (excluding email, $N=43$), file management ($N=40$), and the operating system ($N=40$) (see Fig. 3).

The most common task participants were engaged in at the time of the reported notification was browsing the web (Q6, coded. See Fig. 4. Note: some participants indicated multiple tasks in their descriptions).

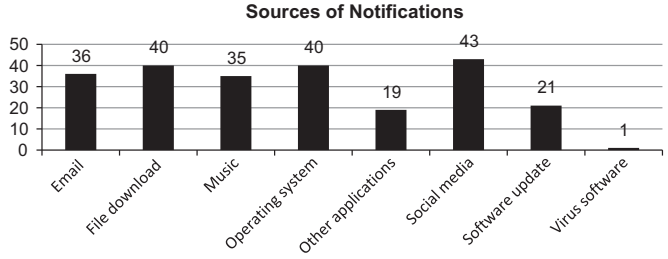


Fig. 3. Sources of interruptive notifications.

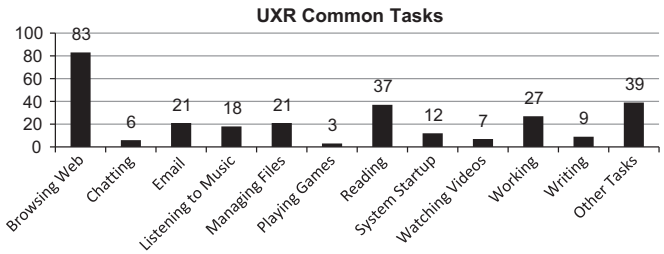


Fig. 4. User tasks at the time of reported notification.

Participants were more likely to not stop their main task to attend to the notification (Q7; Fig. 5).

The notifications reported in the UXR were not usually related to the participants' main task (Q8; Fig. 6).

4.1.2. User experience

Participants were twice as likely to describe notification experiences with positive words than negative words (Q5, coded; Fig. 7).

The most common positive word used to describe notification experiences was “good” ($n=45$; Fig. 8 and 9). The most common negative word used to describe notification experiences was “annoying” ($n=11$; Fig. 2).

Although participants were three times as likely to want similar future notifications, 25% of participants said wanting a similar future notification would depend on a variety of factors (Q4; Fig. 10).

Overall, participants seemed to have generally positive notification experiences. In the next section, we describe the positive and negative aspects of notifications as they support task management.

4.2. Task management behaviors

Our study design was such that the UXRs provided initial insight to the general notification user experience and the interviews were an opportunity to go into more detail of themes identified in preliminary analysis of the UXR study. This section describes the results of a synthesized analysis of the UXR and interviews with regards to task management behaviors.

Task management, specifically with regards to multi-tasking or rapid task switching, was a common activity participants discussed when describing their notification user experiences (Q7 and Q8). The use of notifications to support task management was also explored during the interviews. In the UXR, participants were asked if they stopped their task in order to respond to the notification (Q7: Yes=98, No=128) and if the notification was related to the task it interrupted (Q8: Related=46, Not Related=177). In addition to the survey response, participants were asked to explain in an open-text response. The responses to these two questions were used to provide context to the qualitative responses from the UXR and supplement analysis of the interviews.

The following sections describe how notifications can support multitasking in knowledge-intensive work environments. Through analysis of participant UXR responses and interviews, we were able to identify ways notifications support multitasking, task prioritization, task management, and task disruption management strategies. We identify participant quotes from the UXR responses with (UXR) and from the interviews with (INT) as well as identify the participant (ID).

4.2.1. Notifications and multitasking

Multitasking is the switching between two or more tasks related to the same goal or different goals. Notifications are a cue to help identify when a user might want to switch to a different task.

Notifications gave users an opportunity to manage multiple tasks at the same time:

"For some cases, [notifications] support my workflow quite well." (INT2)

The mere existence of interruptive notifications implies multitasking. In some cases, notifications from one task, even if a passive background task, interrupted an active focus task. In other cases, notifications introduced new tasks. While this is an expected characteristic of the context, we found strong confirmation of this in our data. Previous research (Gillie and Broadbent, 1989; Mark et al., 2005, 2008; Ardissono et al., 2009) found conflicting evidence regarding the impact of the relationship of the main task and the interruption task. While some of this research (Mark et al., 2005; Ardissono et al., 2009) found that interruption tasks that were similar and related to the main task created less of a disruption, other studies (Mark et al., 2008) found no difference in impact. Instead of simply establishing the existence of a relationship between the main task and interruption task, we identified four specific types of multitasking behaviors where the relationship between the main and the interruption tasks varied. While we did not measure the extent of the interruption for each of these, we describe the different user behaviors surrounding these relationships. Participants in our study may have worked on several unrelated tasks at the same time, create a filler task while a more important task was completing in the background, temporarily switch attention to complete something brief, or work on a common goal while switching between multiple, related tasks.

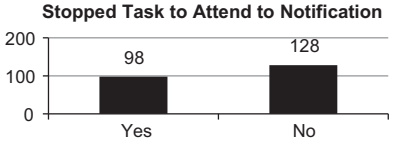


Fig. 5. Frequency of participants who stopped their main task in order to attend to the reported notification.

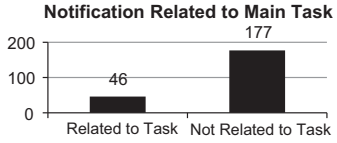


Fig. 6. Frequency of notifications related to participants' main task.

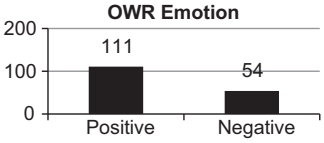


Fig. 7. Frequency of positive and negative words used to describe the overall notification user experience in the One Word Response.

The first type of multitasking behavior was **task overlapping**. The task environment participants described often included overlapping parallel tasks, that is, participants would work on several unrelated tasks at different stages at the same time. Participants capitalized on notifications to manage multiple tasks or activities at the same time.

> "Because with KDE's notifications you can do other things while processes are running or you have the information." (UXR75)

> "I have to be notified when a file operation finished. Meanwhile I can do other tasks." (UXR205)

> "I don't have to keep an eye on one or more IRC channels and can do other stuff." (UXR211)

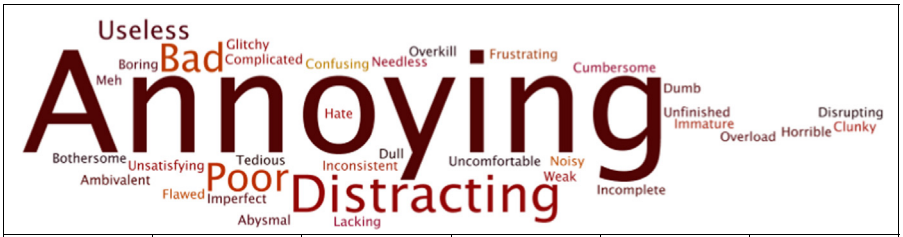
> "I'll have a meeting lined up on [IRC] for a given time of the day, and the person is supposed to message me when they are ready. I'll keep busy doing other things until they ping me and when they do (which I notice via a notification somewhere) then I switch to the meeting." (INT1)

> "Sometimes during chat conversations with people who have long response times, I do something else while waiting for an answer." (INT3)

> "I click on the download, tell it where to save the file, do other stuff while the systray icon is showing the download. Then I get a notification saying it's done, meaning I'm ready to watch Game of Thrones." (INT4)

Good	45	Positive	2	Fair	1
Great	6	Simple	2	Fast	1
Useful	5	Smooth	2	Like	1
Awesome	4	:)	1	Love	1
Excellent	4	Accurate	1	Reasonable	1
Helpful	4	Adequate	1	Relief	1
Satisfactory	4	Brilliant	1	Sharp	1
Cool	3	Convenient	1	Shiny	1
Decent	3	Correct	1	Smart	1
Fine	3	Easy	1	Tolerable	1
Nice	3	Enlightening	1	Unobtrusive	1
Pleasant	2				

Fig. 8. Frequency of Positive One Word Responses used to describe the overall reported experience.



Annoying	11	Confusing	1	Horrible	1
Distracting	4	Cumbersome	1	Immature	1
Bad	3	Disrupting	1	Imperfect	1
Abysmal	1	Dull	1	Incomplete	1
Ambivalent	1	Dumb	1	Inconsistent	1
Boring	1	Flawed	1	Lacking	1
Bothersome	1	Frustrating	1	Meh	1
Clunky	1	Glitchy	1	Needless	1
Complicated	1	Hate	1	Noisy	1

Fig. 9. Frequency of Negative One Word Responses used to describe the overall reported experience.

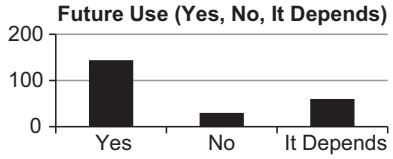


Fig. 10. Frequency of responses for wanting similar future notifications.

Sometimes overlapped tasks were purposeful for planning a future task or activity.

> "I started the operation in preparation for something I wanted to work on a bit later." (UXR8)

The second type of multitasking behavior was **task filling** with less important tasks. In some cases, a continuous or dependent task, such as a file download or code compiling, was often overlapped with a filler task the person would work on while waiting for the main task to complete. These filler tasks were often stand-alone activities of little consequence compared to the main task, such as reading a web page while downloading a file needed for the main task.

> "I was browsing in the meantime letting the files copy." (UXR27)

> "Because I was making time surfing the net until the script finished." (UXR197)

> "I… browsed Reddit because I didn't want to start doing something more sophisticated that would keep me from finishing preparing the tea." (He was waiting for a timer notification indicating the tea was steeped.) (UXR207)

Tasks with large gaps between activities, such as asynchronous chatting or file downloads, were the most common activities to be filled with a filler task. Web browsing tasks, such as reading news or blogs, were a common filler task.

The third type of multitasking behavior was **temporary task switching**. Temporary switches of focus, such as to read the notification message, are not necessary true task switches. However, these types of notifications help plan tasks through the information they provide. Sometimes these notifications act as directives to new tasks or simply provide information about a change in a passive background process.

> "[Notifications] let me know when someone sends me a message and lets me read it without switching from what I'm currently working on." (UXR3)

The fourth type of multitasking behavior was **goal-related task switching**. There were examples of multitasking with a single activity or goal in mind, such as switching between related but different tasks that contribute to the same end goal.

> "[The notification is] part of the flow in this case. I was chatting while waiting for a command output (network was slow), but I regard the chat and command as the same activity." (UXR162)

> "[I was copying a] show that I wanted to watch while reading the website of said show." (UXR176)

These four multitasking behaviors show how notifications are an essential part of a user's task management strategy, and that notifications have greater utility beyond just delivering information.

4.2.2. *Task prioritization*

Task prioritization is the selection of an active task based on its relative importance to other possible tasks. The nature of notifications presents a task management and prioritization problem to the user. When a notification was received, participants had to decide if their current task or the notification task was more important. Notifications helped participants prioritize which task to attend to immediately or later.

> "[I] use many programs at once and use the notifications to alert the most relevant of my applications. Since I can only address [one] application [at a time]." (UXR87)

> "[The notification] helps me know in short who is saying what to me and gives me the option to respond now or delay till later." (UXR233)

Task prioritization was a combination of multiple factors that helped participants decide if they should attend to the notification now or later. Participants described three types of task prioritization characteristics.

The first type of task prioritization factor was the **notification source**. For example in a social notification, the sender of the notification is the primary indicator used for prioritization. While earlier work showed the importance of social notifications (Paul et al., 2011a), this observation further refines that result. Not all social contacts are of equal importance and participants described

using who the notification came from as a way of prioritizing task switching.

> "I try to ignore my messaging apps while working, and those notifications let me know when I need to stop what I'm doing to take care of something." (INT1)

> "It's useful to me when notifications carry some information about the origin (e.g. the name of the person trying to reach me) so I can decide whether it's worth interrupting my current activity (i.e. reacting to the notification by switching to the app where I can fold-up or see the entire conversation)" (INT6)

The second type of task prioritization factor was the **notification purpose**. Work by Avrahami et al. (2008) showed that the content of notifications is important in determining user reactions to a notification and our results further explore the impact of a specific characteristic of notification content. Notifications provide feedback on actions and errors. Participants were most interested in notifications that alerted them of unexpected behavior, especially if it was about something urgent or critical. It was important that the confirmation or error message provided useful information that the user could react to; otherwise, the notification was a distraction.

> "I don't need information that something was successful, in this particular case it's not an important message, an error would be important." (UXR130)

> "[Receiving a confirmation notification is good] for every process I have started manually, that can easily fail and where I don't see the result directly (a login process wouldn't need it, as I directly see whether I'm logged in or not)." (UXR224)

The third type of task prioritization factor was the **notification importance**. Urgency was the most common importance characteristics described by participants. Vastenburg et al. (2008) also found urgency to be an important factor for determining the priority of a notification. For example, when the task indicated by the notification was not urgent and could be delayed or attended to at a later time. Interestingly, software and security updates were often lower priority than the current user task and response delayed until a later time.

> "Security updates I'll care about but not right away." (INT4)

However, low priority tasks were never completely ignored. Notification tasks that were lower priority than the current task were often scheduled and responded to at a later time.

> "I didn't have to stop, I got notified about the things I might have to pay attention to (I'm specifically talking about the battery getting low), took it in consideration, keeping it in mind that soon I'll have to act on some of the notifications to be able to keep doing what I am doing." (UXR171)

There are many factors that contribute to a notifications utility. Where the notification came from, what the notification is about, and how important the notification information is are three factors that were observed in our studies.

4.2.3. Task administration with notifications

Task administration is the process of controlling the focus and flow of multiple tasks. Notifications support task management by providing cues to users to indicate when they can switch to a new task. Task switching cues were a way for participants to maximize work efficiency. Rapid task switching was accomplished by participants using notifications as cues to signal when they can or should switch tasks. Notifications can indicate when it is possible to switch to a new or ongoing task.

> "[Notifications] inform me at which point (time) I can start browsing, fetch e-mails, etc." (UXR192)

> "It's useful to know when some job is finished so I can start another." (UXR201)

> "I like being told 'this torrent is done' as then I can go and watch whatever I was downloading" (INT1)

Participants described three types of task management behaviors that relied on task switching cues. The first type of task management behavior was **task planning**. While a task is ongoing, participants found the task progress indicator (part of the KDE notification system) important to know when to expect a notification and to help them with planning their next tasks.

> "I wanted to check the speed [of the download] and look at the notification." (UXR35)

Notifications and progress indicators were also useful for helping participants plan activities. This was especially true for tasks that were suspended to the background for a long period of time, such as a large file download or code compiling. At the same time, some participants felt notifications were unnecessary for activities that only took a short period of time and were still in their focus.

> "This notification should only appear if I am not looking to Konsole [command line interface] and the script has taken a long time to finish." (UXR197)

> "When the music player window is open in the same screen, the notification [about the music player and information in view] is not interesting." (UXR212)

Liu et al. (2014) also found that progress indicators were critical to task resumption and overall user satisfaction of the system.

The second type of task management behavior was **task scheduling**. As previously discussed, participants engaged in a variety of multitasking behaviors that involved switching between tasks in an efficient way by task overlapping and task filling. These multitasking behaviors combined with task planning and prioritization helped participants to schedule the order of tasks. The KDE notification system has a feature that allowed participants to view a history of recent notifications. This history aided in task scheduling and helped participants remember what tasks they delayed until later.

> "I like the ability to get [notifications] back if I missed them." (INT5)

> "I like that I can see [the notification] again and again. It doesn't go away once opened. For me this is good." (INT13)

The third type of task management behavior was **task response**. If a notification had an obvious response, participants appreciated when there were action buttons on the notification to help them immediately switch to the relevant task. This functionality minimized the need for context switching and allowed participants to more quickly return to their main task, thus minimizing disruptions caused by the interruption.

> "It provides quick access to the inserted device to choose an action" (UXR6)

> "[The notification] is a good shortcut to the package manager" (UXR82)

> "If I plugin something it's something I want to work with now, so it's supporting the task at hand" (INT2)

Task switching cues were almost always paired with a filling task or a high priority task. This makes sense since there is not much incentive for a user to forcibly switch a task unless it is part of a multitasking workflow or a high priority task.

4.2.4. Managing task disruptions

Task disruptions are interruptions to tasks that affect the main task in a negative way. While many participants experienced constant and regular interruptions, few were actual disruptions to their tasks. Good notifications had some purpose or use, even if it was lower priority than the current task, because they helped support awareness of other services.

> "[If I didn't have a notification] I would need to check the application always… and thus I'd need to interrupt my work regularly." (INT5)

Although less common, there were cases in which participants received a notification at such a bad time that it was more than an interruption—it became a disruption.

> "When I'm working on something, or reading something important that requires attention - no I don't want the notification." (UXR44)

> "If I'm showing an Impress [open source slide desk like PowerPoint] presentation I wouldn't like it." (UXR53)

> "Usually it's helpful knowing that everything I was waiting for to download has arrived. This time I was recording a video interview at the time, so it was *not* helpful." (UXR88)

Participants described two task disruption management strategies. The first task disruption management strategy required participants to self identify an interruption-sensitive **context of work**. For example, a few heavy multitaskers (frequently software developers) discussed how interruptions unrelated to their current task did affect their work negatively.

> "After muttering under my breath about how I hate being interrupted all the time… then I usually put aside what I'm doing and move my attention to the communication. That I don't have a very good workflow now. Usually I just abandon what I'm doing temporarily and hope I'll come back to it eventually." (INT1)

Some participants who were able to identify a context of work that was sensitive to interruptions, described establishing a "no notification zone". While earlier research described user strategies to create low-load points to in the main task to deal with interruptions (Bogunovich, 2011), a feature of the KDE systems allows for a complete blocking of a notification interruption. The KDE notification system has a feature that allows users to easily temporarily disable notifications and later re-enable them. This feature was critical to some participants' notification user experience.

> "Sometimes I disable these messages, depending on what I'm doing. If doing random stuff I like to see them, but not if I want to [concentrate]." (UXR76)

The importance of context of work in an interruptive environment was also identified by Mark et al. (2005) and Ardissono et al. (2009).

The second type of task disruption management strategy was **task stabilization**. Task stabilization was defined by Iqbal and Horvitz (2007) as the completion of a subtask of the main task in order to prepare to return to it after an interruption. Some participants gave examples of notifications that fell in the middle of an urgent-to-ignore scale and dealt with them by first stabilizing their task before switching at their convenience.

> "There's that valley between the high urgency [contacts] and the interesting-because-they-are-unexpected [contacts] – anything that falls in between those peaks I might let sit for a while if I, say, want to finish writing something down or thinking something through first." (INT6)

Task stabilization behavior was also observed by Cutrell et al. (2001), Czerwinski et al. (2004), Parnin and DeLine (2010), and Andrews et al. (2009).

5. Implications

There are many implications that can be derived from the findings from these two user studies. First, we synthesize the ways in which notifications can support task management. Next, we discuss an alternate view of the results that considers the influence of task and notification factors on the notification user experience. Finally, we present design guidelines for interruptive notification systems that are informed by the positive and negative experiences described by our participants.

5.1. Notifications in support of task management

Through the user experience reports and detailed follow-up interviews, we were able to develop a rich understanding of how people use notifications to support task management. Participants described experiences with their own notifications on their own computers. The following is a summary of the task management behaviors reported in the Results (Section 4).

Multitasking is the switching between two or more tasks related to the same goal or different goals. Notifications served as a cue to identify when a participant might want to switch to a different task. Participants in our research exhibited four types of multitasking behaviors.

We observed several different types of multitasking scenarios in our data. *Task overlapping* occurs when users switch between several unrelated tasks at the same time with the intent to eventually achieve two task goals. *Task filling* occurs when users fill temporarily suspended main tasks with tasks of less importance. It is similar to overlapping except that the filler tasks are usually unimportant tasks that discontinue once the user receives a notification that the main task is complete. *Temporary task switching* occurs when users utilize information in the environment to gain awareness and plan tasks without suspending the main task. This is not necessarily a true task switch but a temporary shift in focus. *Goal-related task switching* occurs when users switch between tasks that are related to the same task goal. For example, participants described switching from a file download task to a reading a webpage about the file while waiting for the download to complete.

Our participants took advantage of notifications to manage multitasking via several different strategies. **Task prioritization** is the selection of an active task based on its relative importance to other possible tasks. Participants in our research considered three types of information for task prioritization. *Notification source* is evaluated to determine the importance of the sender of the interruption in relation to the current task. Paul et al. (2011a)also observed how the social contacts were an important factor for determining the priority of a notification. *Notification purpose* is evaluated to determine the importance of the notification information in relation to the current task. *Notification importance* is a function of the importance of the current task compared to the importance of the notification, such as the case of urgency of the notification message. Vastenburg et al. (2008) also observed urgency as an important factor for determining the priority of a notification.

Task administration is the process of controlling the focus and flow of multiple tasks. Participants in our research conducted three types of administration activities. *Task planning* is when users expect a new task or task switch and try to control the order in which task switching occurs, such as using progress indicators to assist with task planning decisions. Liu et al. (2014) also found that progress indicators were critical to task resumption and overall user satisfaction of the system. *Task scheduling* is when users make decisions about when to switch between tasks to optimize efficiency. *Task response* is when users stop or suspend their current task in order to attend to a new task.

Task disruption required users to minimize the impact of an interruption on a task and reduce the potential for disruption. Participants in our research described two types of task disruption scenarios. *Context of work* is important for understanding the effects and consequences of an interruption, such as if an interruption is related to the current task. The importance of the context of work within an interruptive environment was also discussed by Mark et al. (2005) and Ardissono et al. (2009). *Task stabilization* is when users receive an interruptive notification and they prepare their current task for a temporary switch in focus. This behavior was also observed by Cutrell et al. (2001), Czerwinski et al. (2004), Iqbal and Horvitz (2007), Parnin and DeLine (2010), and Andrews et al. (2009).

Although many of these behaviors have been reported in the prior research that we outlined earlier, our work enhances these findings in at least two specific ways. First, it enhances the ecological validity of the replicated results by providing a naturalistic context to behaviors that had only been observed in artificial laboratory settings or in very limited context in a field study. Second, it deepens these findings by situating them within the cultural context of knowledge workers who are using the same notification system for similar work tasks, but in different personal work environments.

5.2. Notification-task management user experience

In this research, we have described how interruptive notifications support task management in a variety of ways. Understanding the interplay between various factors that influence the user experience of notifications is important for effectively supporting task management using notifications. One way of discussing the results of this research is to map the relationships between important factors in the notification-task management user experience. This map illustrates the positive and negative influences of contextual factors on three common usage scenarios (Fig. 11).

The first user experience scenario was **notification relatedness and task salience**. If a notification is related to a task currently in focus, the user experience will be negatively influenced. If the task is in focus, it is likely that the user either deliberately took the action causing the notification or can already see the information the notification is alerting on and a secondary notification is not necessary. All the notification accomplishes is drawing attention away from the current task. The purpose of a notification is to provide awareness of information outside the user's attention. However, if a notification is related to a current task that is not currently in focus, the user experience will be positively influenced. The notification is providing valuable awareness of a current task that may be ongoing in the background, possibly providing a task switching cue in support of task management.

The second user experience scenario was **notification response**. The presence or absence of action buttons on notifications that have a potential response influences the notification user experience. Notifications that have a potential response action or reaction and provide supporting response buttons have a positive influence on the user experience. Notifications that do not provide action buttons have a negative influence on the user experience. Action buttons allow users to quickly react to task switching cues with limited disruption to their workflow. Notifications without action buttons require users to suspend their task resulting in a task disruption, potentially making it more difficult to return to the original task.

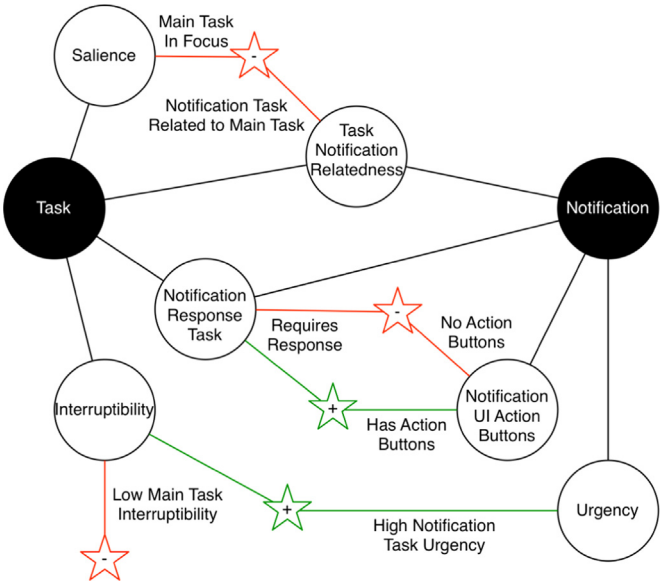


Fig. 11. Map of influence factors in the notification user experience.

The third user experience scenario was **task interruptibility**. The interruptibility of a task, or the consequence of interrupting a task, was a strong influence on the notification user experience. Notifications received during a task with low interruptibility will negatively influence the user experience. Task with low interruptibility described in our study included full-screen movies, slide-show presentations, and screen captures. Exceptions to this are notifications with high urgency; for example, if the user is watching a movie and his laptop battery is running low, interrupting his movie to remind him to find a power source would be acceptable.

Knowledge of the positive or negative influences of a task or notification factor on the user experience provided valuable insight for creating guidelines for designing effective and enjoyable notification systems.

5.3. Notification system design guidelines

The results of this research can translated into design guidelines that can help system designers apply this research to their own interruptive notification systems. Although these design guidelines are derived from experiences with the KDE notification system, many of these principles are relevant to any desktop notification system. Features that are unique to KDE, such as an integrated task progress indicator and a history of past notifications, provide evidence for the benefit of such features for task management and a successful interruptive notification system.

5.3.1. Use the notification system to provide status of tasks that take a long time to complete

As described in Section 4.2.1 Notifications and Multitasking, participants used the notification system to check the process of a task as a way to help plan their next task and filler tasks. This feature of the KDE notification system – managing the status of notification tasks in one place – received positive feedback from users. While newer versions of Microsoft Windows and Apple Mac

OS X have better integrated notification systems than in previous versions, they have yet to provide the level of integration that KDE's notification system has achieved. Additionally, KDE notifications utilize the open source D-Bus messaging standard[3] that allows KDE applications in non-KDE environments to send notifications using the native system's design and behavior rules, such as Gnome[4] or Unity.[5]

5.3.2. Only interrupt the user when something unexpected happens. Do not confirm expected results

Several of the reported notification experiences that were described as having an impact included notifications that confirmed behavior that was expected to be successful by default, such as sending an email (as described in Section 4.2.3 Task Administration with Notifications). The utility of a notification is a function of the cost of interruption and value of the information it delivers. If that information has little or no value, then the notification has an unbalanced cost to the user. KDE allows users to extensively configure the behavior of notifications. However, the burden of self-configuration often leads to users not changing defaults as often as the developer may expect. As a result, many developers opt to turn notifications on by default, leading to many unnecessary notifications. The experiences described in Section 4.2.4 Managing Task Disruptions provide some examples that would benefit from customization for a particular context. Better baseline configurations that treat the KDE system as a whole, rather than individual applications, may help system designers make better decisions about default notification configurations. A number of factors that contribute to notification prioritization and valuation are described in Section 4.2.2 Task Prioritization.

5.3.3. Provide ways to easily switch to notification related tasks to minimize disruption to main task

Section 4.2.3 Task Administration with Notifications describes some of the ways that participants utilized action buttons on notifications. These buttons had a direct impact on the user experience of that notification event. If a notification supported an obvious and quick response and a button that supported that action did not exist or was broken, participants expressed great displeasure. The action buttons were a convenient way of switching to the interruptive task in the context of the interruption. The action buttons minimized the cost of disruption by supporting a quick response. Without action buttons, users of a system would have to open the notifying application or service manually, a much more involved and disruptive task than simply clicking on a salient button. Users may be more likely to delay a response or ignore a notification if action buttons are not supported.

5.3.4. Do not notify a user of information that Is already in focus

Notifications related to new or background tasks provide more value than tasks already in focus. Several participants described cases of when an application sent a notification while the application was in focus, making the notification redundant and an unnecessary distraction (as described in Section 4.2.3 Task Administration with Notifications with additional examples in Section 4.2.4 Managing Task Disruptions). An application or service should be smart enough to know if its user interface is in focus. When in focus, application should support contextual alerts and notifications, rather than rely on the global notification system. Only use a notification if it can provide useful information to the user, otherwise the value and utility of the notification system is decreased.

6. Conclusions

While notifications are a popular HCI research topic, most of the research has focused on the negative effects of notifications on users. Our work looks at notifications from more neutral perspective, understanding that notifications also provide a meaningful benefit to users. Our field studies helped us to better understand how notifications are used to support task management.

The novelty of our work lies in three areas. First, we used a specific experience sampling method called a User Experience Report that provided unique insights into the interruptive notification user experience in the user's natural environment. These grounded results may not have been accessible using other data collection methods. Second, the UXR method revealed the user's actual notifications in the user's natural environment rather than artificially augmenting the environment with simulated notification events. While similar methods have previously been used to study interruptions, the UXR method is the only one to focus on user experience, use a model for notification context, and study naturally occurring interruptive notifications. Thus, we feel that our data is particularly valuable in confirming findings of earlier experience sampling studies and for highlighting new patterns of behavior. Third, we study a novel domain and participant group. The KDE open source project is a community of software developers and users with a wide range of technical skills. Their interest and background in technology make them a successful surrogate for a general class of technical knowledge workers. Additionally, the nature of the KDE desktop environment ensures that users all use the same applications, services, and integrated notification system. This consistency of population and technology provides a novel way of controlling independent variables while studying a naturalistic work environment.

Multitasking is an increasingly necessary behavior of knowledge workers in today's information-rich, distributed work environments (such as the type of environment modeled so well by the KDE community). Interruptive notifications provide an important service that helps users maintain an awareness of the status of the multiple tasks they are managing at any given time. While the disruptive nature of notifications is well documented, in this paper we highlight the supporting role that notifications can play in users' task management activities. Understanding how users use notifications to manage multiple tasks in their workflow can help designers create better notification system.

Through our use of the UXR method we gained a rich understanding of the role of interruptive notifications in task management. Notifications can be a very valuable if designed correctly; however, it is easy to overburden users with poorly designed notifications. Our results show that it is important to provide notifications about long tasks or unexpected events, but notifications from tasks that are currently in focus have little benefit and can be disruptive. It is also important for the notification system to be aware of what the user is doing and not interrupt certain types of tasks, such as full screen video conference calls. Notifications should provide an easy way for users to switch to the task in the notification, such as through buttons in the notification interface. Ultimately, a unified notification system where users can easily review recently received notifications will make task management and planning much easier.

Notifications will continue to play an important role in assisting users attending to their varied information services, such as the knowledge workers in our studies. Continuing to investigate the interruptive notification user experience is essential for supporting a future knowledge worker-friendly computing environments.

[3] D-Bus, http://www.freedesktop.org/wiki/Software/dbus
[4] Gnome, http://www.gnome.org
[5] Ubuntu Unity, http://unity.ubuntu.org

There is little doubt that the design of notification systems has an effect on how well users can manage their tasks.

Acknowledgements

We would like to thank Carolyn Seaman, and Henry Emurian from the University of Maryland Baltimore County, and Emile Morse from the National Institute for Standards and Technology for their contributions to this research.

References

Andrews, A.E., Ratwani, R.M., Trafton, J.G., 2009. The effect of alert type to an interruption on primary task resumption. In: Proceedings of HFES 2009.
Ardissono, L., Bosio, G., Goy, A., Petrone, G., 2009. Context-aware notification management in an integrated collaborative environment. In: Proceedings of UMAP 2009, pp. 21–30.
Avrahami, D., Fussell, S.R., Hudson, S.E., 2008. IM waiting: timing and responsiveness in semi-synchronous communication. In: Proceedings of ACM CSCW 2008, pp. 285–294.
Beaudry, K., Pinsonneault, A., 2010. The other side of acceptance: studying the direct and indirect effects of emotions on information technology use. MIS Quart. 34 (4), 689–710 (2010).
Berry, B., 2003. Adapting heuristics for notification systems. In: Proceedings of ACM SE 2003, pp. 44–149.
Birnbaum, L., Bareiss, R., Hinrichs, T., Johnson, C., 1998. Interface design based on standardized task models. In: Proceedings of ACM IUI 1998, pp. 65–72.
Bogunovich, P., Salvucci, D., 2011. The effects of time constraints on user behavior for deferrable interruptions. In: Proceedings of ACM CHI 2011, pp.3123–3126.
Cades, D.M., Trafton, J.G., Boehm-Davis, D.A., Monk, C.A., 2007. Does the does the difficulty of an interruption affect our ability to resume? In: Proceedings of HFES 2007.
Cadiz, J.J., Venolia, G., Jancke, G., Gupta, A., 2002. Designing and deploying an information awareness interface. In: Proceedings of ACM CSCW 2002, pp. 314–323.
Consolvo, S., Walker, M., 2003. Using the experience sampling method to evaluate Ubicomp applications. IEEE Persuas. Comput. 2 (2), 24–31 (April 2003).
Cutrell, E., Czerwinski, M., Horvitz, E., 2001. Notification, disruption, and memory: messaging interruptions on memory and performance. In: Proceedings of ACM CHI 2001, pp. 263–269.
Czerwinski, M., Horvitz, E., Wilhite, S., 2004. A diary study of task switching and interruptions. In: Proceedings of ACM CHI 2004, pp. 175–182.
Fogarty, J., Ko, A.J., Aung, H.H., Golden, E., Tang, K.P., Hudson, S.E., 2005. Examining task engagement in sensor-based statistical models of human interruptibility. In: Proceedings of ACM CHI 2005, pp. 331–340.
Forlizzi, J., Battarbee, K., 2004. Understanding experience in interactive systems. In: Proceedings of ACM DIS 2004, pp. 261–268.
Free/Libre/Open Source Software: Policy Support, 2006. Gender: integrated findings report. Deliverable D16. ⟨http://flosspols.org/deliverables/FLOSSPOLS-D16-Gender_Integrated_Report_of_Findings.pdf⟩. Last accessed January 2013.
Giveska, S., Sibert, J., 2004. Empirical validation of a computer-mediated coordination of interruption. In: Proceedings of ACM OZ-CHI 2004.
Gillie, T., Broadbent, D., 1989. What makes interruptions disruptive? A study of length, similarity, and complexity. Psychol. Res. 50, 243–250.
Gluck, J., Bunt, A., McGrenere, J., 2007. Matching attentional draw with utility in interruption. In: Proceedings of ACM CHI 2007, pp. 41–50.
González, V.M., Mark, G., 2004. "Constant, constant, multi-tasking craziness": managing multiple working spheres. In: Proceedings of ACM CHI 2004, pp. 113–120.
Gould, S.J.J., Brumby, D.P., Cox, A.L., González, V.M., Salvucci, D.D., Tattgen, N.A., 2012. Multitasking and Interruptions: a SIG on bridging the gap between research on the micro and macro worlds. In: Proceedings of ACM CHI 2012, Extended Abstracts, pp. 1189–1192.
Hektner, J.M., Schmidt, J.A., Csikszentmihalyi, M., 2007. Experience Sampling Method: Measuring the Quality of Everyday Life. Sage Publications, Thousand Oaks p. 2007.
Hudson, J.M., Christensen, J., Kellogg, W.A., Erickson, T., 2002. "I'd be overwhelmed, but it's just one more thing to do:" availability and interruption in research management. In: Proceedings of ACM CHI 2002, pp. 97–104.
Iqbal, S.T., Horvitz, E., 2007. Disruption and recovery of computing tasks: field study, analysis, and directions. In: Proceedings of ACM CHI 2007, pp. 677–686.
Iqbal, S.T., Bailey, B.P., 2008. Effects of intelligent notification management on users and their tasks. In: Proceedings of ACM CHI 2008, pp. 93–102.
Iqbal, S.T., Horvitz, E., 2010. Notifications and Awareness: a field study of alert usage and preferences. In: Proceedings of ACM CSCW 2010, pp. 27–30.
ISO 9241-11, 1998. Ergonomics of Human System Integration – Part 11: Guidance on Usability. International Organization for Standards (ISO), Switzerland.
ISO 9241-210, 2009. Ergonomics of Human System Integration – Part 210: Human-Centered Design for Interactive Systems (formerly known as 13407). International Organization for Standards (ISO), Switzerland.
KDE, 2012. Akademy 2012 survey results. Private communication from KDE Akademy program committee. Received August 2012.
Korhonen, H., Arrasvuouri, J., Väänänen-Vainio-Mattila, K., 2010a. Let users tell the story: evaluating user experience with experience reports. In: Proceedings of ACM CHI 2010, Extended Abstracts, pp. 4051–4056.
Korhonen, H., Arrasvuouri, J., Väänänen-Vainio-Mattila, K., 2010b. Analysing user experience of personal mobile products through contextual factors. In: Proceedings of ACM MUM 2010.
Lakhani, K.R., Wolf, R.G., 2005. Why hackers do what they do: understanding motivation and effort in free/open source software projects. In: Feller, J., Fitzergald, B., Hissam, S.A., Lakhani, K.R. (Eds.), Perspectives on Free and Open Source Software. MIT Press, Cambridge.
Liu, Y., Jia, Y., Pan, W., Pfaff, M., 2014. Supporting task resumption using visual feedback. In: Proceedings of ACM CSCW 2014, pp. 767–777.
Maglio, P.P., Campbell, C.S., 2000. Tradeoffs in displaying peripheral information. In: Proceedings of ACM CHI 2000, pp. 241–248.
Mark, G., González, V.M., Harris, J., 2005. No task left behind? Examining the nature of fragmented work. In: Proceedings of ACM CHI 2005, pp. 321–330.
Mark, G., Gudith, D., Kocke, U., 2008. The Cost of interrupted work: more speed and stress. In: Proceedings of ACM CHI 2008, pp. 107–110.
Matthews, T., Dey, A.K., Mankoff, J., Carter, S., Rattenbury, T., 2004. A toolkit for managing user attention in peripheral displays. In: Proceedings of ACM UIST 2004, pp. 247–256.
McCrickard, D.S., Chewar, C.M., Somervell, J.P., 2003. A model for notification systems evaluation—assessing user goals for multitasking activity. ACM Trans. Comput. Human Interact. 10 (4), 312–338.
Miller, T., Stasko, J., 2002. Artistically conveying peripheral information with the InfoCanvas. In: Proceedings of ACM AVI 2002, pp. 43–50.
National Science Foundation, 2011. Women, minorities, and persons with disabilities in science and engineering. Digest 2011. ⟨http://www.nsf.gov/statistics/wmpd/pdf/nsf11309.pdf⟩. Last accessed January 2013.
Nielsen, J., 1983. Usability Engineering. Academic Press, Cambridge.
Parnin, C., DeLine, R., 2010. Evaluating Cues for resuming interrupted programming tasks. In: Proceedings of ACM CHI 2010, pp. 93–102.
Paul, C.L., Komlodi, A., Lutters, W., 2011a. Again?!! The emotional experience of social notification interruptions. In: Proceedings of IFIP INTERACT 2011, pp. 471–478.
Paul, C.L., Morse, E., Zhang, A., Choong, Y.Y., Theofanos, M.A., 2011b. A Field study of user behavior and preferences in smartcard authentication. In: Proceedings of IFIP INTERACT 2011, pp. 1–17.
Patton, M.W., 2001. Qualitative Evaluation and Research Methods, 3rd edition Sage Publications, Thousand Oaks p. 2001.
Sanders, G.S., Baron, R.S., 1975. The motivating effects of distraction on task performance. J. Personal. Social Psychol. 33 (6), 956–963.
van Dantzich, M., Robbins, D., Horvitz, E., Czerwinksi, M., 2002. Scope: providing awareness of multiple notifications at a glance. In: Proceedings of ACM AVI 2002, pp. 267–281.
Vastenburg, M.H., Keyson, D.V., de Ridder, H., 2008. Considerate home notification systems: a field study of acceptability of notifications in the home. Person. Ubiquitous Comput. 12, 555–566.