Notification Brokering Algorithm for Smartphones

Mehmet Okatan

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 23.5.2016

Thesis supervisors:

Prof. Antti Oulasvirta

Thesis advisor:

M.Sc. Anna Feit

Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Mehmet Okatan

Title: Notification Brokering Algorithm for Smartphones

Date: 23.5.2016 Language: English Number of pages: 6+53

Department of Communications and Networking

Professorship: Communications and Networking

Supervisor: Prof. Antti Oulasvirta

Advisor: M.Sc. Anna Feit

The widespread usage of smartphones brought many advantages to our daily lives since people are able to handle so many things just by touching a screen. On the other hand, this trend started to cause various problems such as distraction, low task performance, high error rate, emotional distress, annoyance and many others due to spoiling character of the notifications. The increase in the average number of mobile applications and data usage has consolidated the lack of balance between disturbance and communication since users are being bombarded with notifications every day.

This research focuses to provide a new algorithm to optimize the tradeoff between communication and disturbance by prioritizing and filtering the notifications, as well as the modality decision. To achieve these goals, the problem is represented with mathematical formulas and realistic scenarios. To optimize the communication and disturbance tradeoff, techniques such as on-line scheduling, probabilistic modeling and multiple resource theory are used, and graphical user interface is built as well in order to be able to follow the simulation results more effectively.

Results show that it is possible optimize the mentioned tradeoff and maintain the desired balance by re-scheduling the incoming messages, sending important notifications and using the right modalities. Different modality decisions for different user selections give the algorithm the capability to control the total amount of disturbance. Also as a consequence of on-line scheduling, messages with higher priorities always have the greater percentage of being sent but on the other hand some acceptable delays in the notifications are observed.

Keywords: Notifications, Multiple Resource Theory, Scheduling, Context-Aware Computing

Preface

I want to express my deep gratitude to Prof. Antti Oulasvirta for providing me this thesis topic, giving me the chance to work in the User Interfaces group, teaching a lot about how to do science and supporting me during my studies.
Additionally I am very grateful to my advisor M.Sc. Anna Feit for guiding me during this thesis with her great ideas, comments and knowledge.
Moreover, I would like thank my sister, mother and father for supporting me no matter what.

Otaniemi, 23.5.2015

Mehmet Okatan

Contents

Symbols and abbreviations

Symbols

A	auditory
a	ambient
b	constant for notifications sent to the user
c	constant for messages waiting in the queue
C	cognitive
d	constant for messages in recycling bin
f	focal
I	interference
P	probability
R	response
r	number of messages in recycling bin
s	spatial
T	time
u	number of notifications sent to the user
q	number of messages in the queue
V	visual
v	verbal

Operators

abs	absolute value
$\sum_i$	sum over index i

Abbreviations

ADHD	Attention Deficit Hyperactivity Disorder
KLM	Keystroke Level Model
MRT	Multiple Resource Theory

1 Introduction

This thesis investigates the way of reaching to an optimal tradeoff between the degree of communication and disturbance in smartphone notification systems. The aim of the work is to create and implement a new notification brokering algorithm by addressing the problem with mathematical formulas and realistic scenarios.
The evolution of computers and internet technologies brought many different devices to people's daily lives, and smartphones are in the center of our routines. With the increased technology and data usage in smartphones, this device became a vital necessity to our routines. Communicating, entertainment, socializing, business, health and many other fields have become more and more integrated to the functions of smartphones. They started to send users more information to keep them updated and notifications stand at the center of information awareness by informing the user with auditory, visual and haptic signals to get user's attention to the incoming messages [1] .
On the other hand, the world is not completely digitalized and humans are supposed to do different things without using a smartphone, too. People need to involve in other activities as well, such as walking, eating, thinking, driving, sleeping and many other things excluding smartphones. In some of these cases situation awareness and cognitive performance are extremely important to ensure one's safety. There are three fundamental areas considering cognitive ability and performance prediction as an important input, which are situation awareness, mental workload and automation [2] and notification systems are highly related to these concepts.
Since people need both information and situation awareness, the tradeoff between these two should be precisely decided in order to create an effective integration and a smooth transition between smartphone-based activities and others.

1.1 Problem

Existing notification systems do not effectively consider the state of the user and environment in order to balance the degree of communication and disturbance optimally as desired. This issue causes numerous different problems such as distraction, high error rates, missing important information, low multitasking performance and many others.
The degree of communication affects the knowledge of the user significantly and the emergence of smartphones has strengthen the conflict between getting information and being disrupted. With the existing notification agents, if the user wants to be informed about the things more, she/he will probably end up facing unnecessary interruption and distraction since there is a lack of notification prioritization according to the user and environment state. This issue might even get more critical if the user is already busy with an important task whereat attention and cognitive ability is a supreme element in such cases [3].
On the other hand if the user wants to be informed only about the necessary things and does not want to be disturbed much, there is a big possibility of missing an important notification or getting disturbed by an unwanted notification. The time

based resource sharing model of attention claims that even a slight transition among different tasks takes cognitive endeavor in addition to the existing load and the theory emphasizes that an increase in cognitive load makes users more vulnerable to other possible distractors [4].
Last but not least, the modality decision for a notification is mostly the same if user does not change any settings and there are many ways to send a notification via different channels. Some notifications use different combinations of auditory, visual and haptic signal while some of them just pop up silently. A dynamic modality decision could play a great role to make users aware of an incoming notification by causing smaller task interference.
Thinking of previously mentioned factors, there is a need for a mechanism to make users lives easier since they keep being bombarded with notifications.

1.2 Motivation

The reason behind conducting this research is quite similar to any smartphone user since they all experience at least some of the mentioned problems. Let's have a look at this basic scenarios that users went through in order to understand the motivation practically.

- ***Scenario 1***
  User 1 has a deadline and she/he is working hard, at the same time does not want to turn the notifications off because she/he is expecting an important e-mail. But she/he ends up receiving tons of notifications from other applications, too. This situation affects her/his multitasking ability and concentration.
- ***Scenario 2***
  User 2 is sitting in a coffee shop and she/he does not want extreme cases such as receiving all the notifications or turning them off completely. To do this she/he needs to specify everything in settings which is a burden for her/him.

Such scenarios always keep affecting smartphone owners negatively thus the motivation of this work is to:

- Provide a clear scientific perspective of the problem
- Improve user experience with notification systems
- Contribute provide a future cocnept for notification systems

1.3 Objectives

The practical purpose of this work is to build a prototype of a new notification agent based on user and environmental state to save users from the hassle of existing notification systems, such as distraction, task interference or missing important information related to mentioned problems. With the contribution of the prototype, the following bullet points are aimed to be achieved:

- ***Maximizing multitasking and communication***
  No matter what the situation is, obviously the most important objective is to maximize the user's multitasking ability and communication capability as much as possible without causing any troubles.

- ***Optimizing the tradeoff between the degree of disturbance and communication by considering user preferences, user state and environmental state***
  To be able to minimize the disturbance and maximize communication degree, these concepts should be expressed mathematically and optimized by considering scientific background.

- ***Filtering and scheduling incoming messages and pushing the important ones as notifications***
  Since first two objectives are strongly related with the number of sent notifications, filtering and scheduling plays a critical role to acheive an optimal state.

1.4 Scope and Constraints

When a problem about human beings is stated with mathematical formulas, some generalizations and assumptions should be set in order to standardize the problem.

Constraints:

- ***Different perceptual processing and attentional capacity of individuals***
  People are unique in many different ways and one of them is their brain. It is known that human cognitive system has a restricted service capacity and if this capacity is already reached there is no room for other tasks [5].

- ***Psychological factors***
  Psychological disorders of individuals, such as attention deficit hyperactivity disorder (ADHD)or the emotional state of user could affect the perceived disruption and vice versa [4].

- ***Hardware settings of the device***
  The hardware settings of the device could be easily controlled or limited by the user since we have no control over the device itself. For example, the owner of the smartphone could limit the volume by using the buttons which could cause to catch less attention in auditory notifications.

Assumptions:

- ***Interference***
  The disparity of interference caused by different auditory,visual and haptic patterns are not considered. All of the auditory,visual and haptic warnings are assumed to be standard.

2 Background

In this chapter, previously done theoretical works and their applications will be examined to have a clear understanding of the subject matter before moving onto the design process. Although there are a lot of domains to explore, it is always important to see the big picture for the sake of the simplicity and go to the details when needed. The detailed information considering the design process will be given in the upcoming chapters.
In this part, three main areas are considered before starting to design a notification brokering algorithm are:

- Multitasking and interruptability
- Scheduling
- Notification systems

2.1 Multitasking and Interruptability

Before taking any kind of decision about notifications, interrupting the user or modalities; a decent scientific background should be considered to be able to obtain appropriate results. Without any doubt, decision of interrupting a user is the most sensitive judgement of this research since the problem itself is users are either being interrupted too much or too less. To make this critical decision, considering user's current situation is a vital necessity. As L.D. Turner et. al. state in their paper: "Assessing another person's interruptibility prior to interaction with them is a natural human behavior that is generally easily handled by the human brain. However, creating such capability in the context of a machine, so that there is harmonious synchronicity with human behavior, is a significant challenge that has important ramifications for the demands placed upon a user" [6]. As we understand from the quote, transferring some properties of human behavior to a digital system is a complex problem with adaptive/dynamic solution requirements, thus a clear mathematical representation of the problem is essential. These mathematical representations should include the static and dynamic variables to calculate the situation quantitatively. Thinking of these factors, keeping the track of the communication and disturbance as different scores derived from the status of the system will be considered at the implementation.
After understanding the problem and its complexity, possible errors should also be taken into account as well. Scott LeeTiernan et. al. state in their paper about the user trust, it should be noted that mistakes to some acceptable extend is quite possible to happen in such notification brokering software because of the probabilistic structure of the algorithm [7]. In this case, the best strategy would be minimizing the mistakes to an acceptable extent by using statistics and modeling the situation with probabilistic scenarios. For example it could be statistically estimated if the user is going to check the notification, according to the priority of the message, state of user, visual focus attention, and other factors.

In this work, multitasking is roughly identified as the ability to perform more than one task at the same time and it is strongly correlated with interruptability. If the decision for interruption has been already made, it is necessary to do it in an optimal way. In such scenarios, smartphone user should be able to multitask as much as possible if she/he is working on some other task at the moment. Because the intangible definitions of the problem and situation, a quantitative method should be followed precisely. To address these issues mathematically and computationally, the multiple resource model, which calculates the task interference between two tasks performed by the same user at the same time, is published by C.D. Wickens. The model helps to make the decision by providing a numerical value of the interference in situations like: "...when is it better to use voice control than manual control, to use auditory rather than visual displays, or to use spatial graphic, rather than verbal material (e.g. maps vs route lists for delivering navigational instructions)" [8]. To help the decision making process as quoted previously, the theory aims to predict the level of interference and demand of two time-shared tasks by offering a four dimensional model. This four dimensional model consists of processing codes, perceptual modalities, visual channels and stages (perception,cognition and response). As it is possible to get information about the user and environmental state, this computational model will be integrated to the brokering mechanism and going to provide numerical values representing the level of the interference, which makes the decision much easier.

2.2 Scheduling

Scheduling has been subject to optimization issues since decades in computer science, telecommunications, automation, operations management and many others. "Scheduling is the allocation of shared resources over time to competing activities. Emphasis has been on investigating machine scheduling problems where jobs represent activities and machines represent resources; each machine can process at most one job at a time" [9]. The logic begind this quote could be adapted to similar scheduling problems, such as notification scheduling, by simply replacing the jobs with incoming messages and machines with human brain.
To apply a scheduling mechanism to this specific case, enough information should be gathered in order to know what kind of scheduling should be done before moving to the design phase. As anyone can predict, incoming messages and their time are not exactly known before they arrive, which makes scheduling more challenging due to incomplete information. That means some decisions should be made without knowing the complete situation and this type of scenarios require online scheduling. There are many different algorithms to apply online scheduling and in this thesis work, none of them are exactly used but Shortest-Remain-Processing-Time (SRPT) has inspired me to develop another scheduling mechanism. In 2001, M. Harchol-Balther et. al. published a paper about applying SRPT scheduling algorithm to web servers to explain if it is possible to reduce the response time of requests at a web server by changing the order of requests. [10]. As a result of the research, they succeed to minimize the response time significantly without having any unwanted impacts on

throughput and CPU utilization by building a queueing mechanism according to the remaining processing time. Applying this technique to minimize other factors is possible and in this work this method will be used to minimize disturbance and maximize communication by queueing incoming messages according to their priority.

2.3 Notification Systems

To achieve the goals of this thesis, a practical background is needed as much as the theoretical background. Since this work should also be applicable to realistic scenarios, existing smartphone notification mechanisms and statistics about smartphone notifications should be examined.

2.3.1 Statistics

Although there are many different statistics available about notification systems and mobile application, not all of them are relevant to this project. After reviewing a lot of papers and filtering the related ones, the information and statistics below will enlighten the methodology and system design.
In 2014, A.S. Shirazi et. al. made an assessment of mobile notifications in a large scale to find what sort of notifications users like and dislike in order to offer designers a decent guideline. One of the most important finding is that almost 50 percent of interactions with the notification happen in the first 30 seconds, which means users react to notifications quite fast because of the disruptive nature of the mobile notifications [11]. Figure 2.1 gives a good idea of the mobile application categories since the study is made with more than 40000 users.

	Number of			
Category	**Users**	**Apps**	**Click times**	**Ratings**
messenger	29,627	10	2,508,203	883
voice & messenger	27,768	9	335,530	470
mail	26,120	8	781,502	610
social	22,173	12	501,159	382
calendar	12,292	5	18,990	305
alarm clock	11,849	7	18,678	50
music	15,366	16	47,786	114
game	3,046	6	43,206	41
market	29,326	6	192,357	156
reader/news	3,958	7	11,036	50
utility	13,202	27	29,990	105
tool	11,494	18	33,352	42
system	27,269	21	229,067	304
other	13,511	21	44,370	124
overall	37,233	173	4,795,226	3,636

Figure 2.1: Application categories and their properties [11]

In another study by M. Pielot et. al., an important detail is pointed out: "4837 (70.6%) of notifications were first viewed through the notification drawer, the remaining 1966 (28.7%) by directly opening the app" and continues "When notifications arrived, the screen had been off in 69.2% of the cases. Given how fast people attended to notifications, this indicates that notifications often triggered interaction with the phone" [12]. These quotations give us a pretty important idea: People view most of the incoming notifications quite fast and this ratio is strongly correlated with the focal attention. This information will play a big role while designing the system in terms of estimating user's probability of viewing a notification.
Another paper published in 2016 claims that user's receptivity about notifications is shaped by the application itself and the meaning of the notification. This view is also supported by the questionnaire they made, demonstrated in Figure 2.2, and the most important reasons to click a notification are about the content and application of the notification or user's status [1].

Option	**Count (%)**
Sender is important	31.546
The content is important	27.129
The content is urgent	14.511
The content is useful	31.546
I was waiting for this notification	15.773
The action demanded by the sender does not require a lot of effort	20.189
At this moment, I was free	37.224

Figure 2.2: Reasons Why Users Participate to Notifications [1]

By taking these statistics into account we can say that the type of application and the state of the user play a very big role in user's interest to view or participate in the notification, thus they should be included in the system in order to create a realistic simulation environment.

2.3.2 Offered Notification Modalities

As every smartphone user knows, almost every device and operating system offer the same channels and ways of showing notifications. In today's technology, the smartphone itself offers auditory, visual and haptic notifications.
Auditory notifications are deliberate to catch the user attention as much as they can by propagating sound waves since this type of notifications consider noisy environments as well and they demand straight interest of participant [13]. Also there are many different type of melodies, warning sounds and beeping patterns to give user more information by only using auditory channels. Haptic notifications (vibrations) are also widely used to warn the user since human beings interact with their surroundings without using any visual cues [14]. The same approach in auditory notifications is

valid for vibrations, too. Haptic interaction could be made in different ways such as using different vibration intervals and patterns or using a different hardware which might produce stronger or weaker vibrations. Another important notification type is visual notification and there are very diverse visual interaction techniques since the design space is relatively large. Even a different screen size might get more attention because of its illumination. Additionally some devices offer an integrated LED light with different colors to warn users while some of the devices do not have them at all. Last but not least, with the emergence of wearable technologies such as smart watches and Bluetooth headsets and many other interaction techniques are getting more and more popular but they are not considered in this work to protect the simplicity and focus on the main aspect.

2.3.3 Notifications in iOS and Android

After considering the theoretical work, the practical applications should also be taken into account and interpreted in the context of mobile area. In this part, two most common operation systems, iOS and Android will be compared.

Notifications While Smartphone is Locked

Different operating systems offer different styles for displaying notifications. In the latest versions of iOS and Android, lock screen notifications look quite similar with a major difference. Since Android groups the incoming notifications according to the application and gives offers better usability, iOS shows all of the incoming notifications on the screen and offers more detailed information without unlocking the screen. Additionally it is beneficial to remind that older versions of these operation systems could offer slight differences.

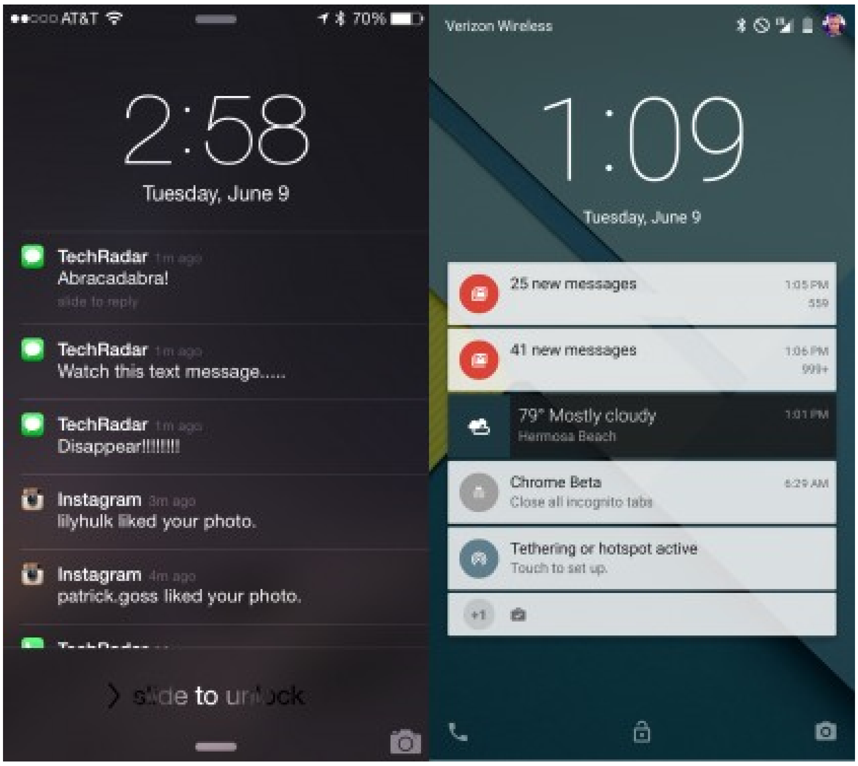

Figure 2.3: Lockscreen notification screenshots of iOS (left) and Android (right)

Notifications While Smartphone is Unlocked

Notifications may arrive while user is using the device as well and manufacturers developed different strategies to catch user's attention because they know that user's visual attention is focal. When a notification arrives, both Android and iOS marks the application symbol with a red bubble which indicates the number of incoming notification until the user attends. But notifications do not always silently appear as a bubble, sometimes they pop up as banner or alert as shown in Figure 2.4 and Figure 2.5

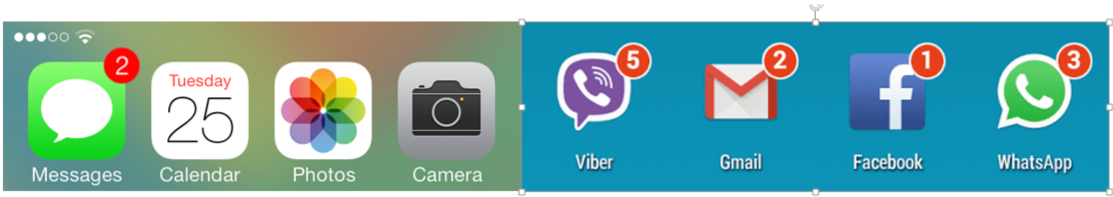

Figure 2.4: Notification bubble screenshots of iOS (left) and Android (right)

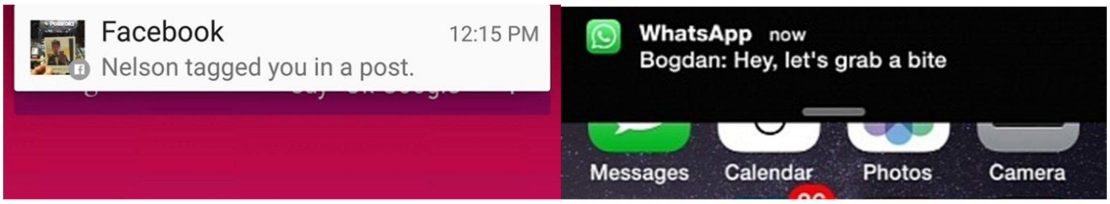

Figure 2.5: Notification banner screenshots of iOS (left) and Android (right)

As seen from the examples there are many different ways of shoving notifications visually and they might vary from device to device but the main concept and techniques are quite similar except some differences.

Do Not Disturb Me Options and Notification Customizing

With the increase in the number of used applications and mobile internet, many people are suffering from getting bombarded with notifications. To find a solution to this problem, both Android and iOS have a "do not disturb me" modes with similar settings.

Android allows users to select their mode according to their wishes with a limited elasticity. User could select "total silence" to silence all the notifications, "alarms only" to allow only alarms to warn, or "priority only" to allow notifications from prioritized applications. User also is allowed to use this mode as scheduled between certain times. On the other hand, iOS does not offer any customization but scheduling the mode. Additionally both iOS and Android users are able to customize which applications will be able to send notifications.

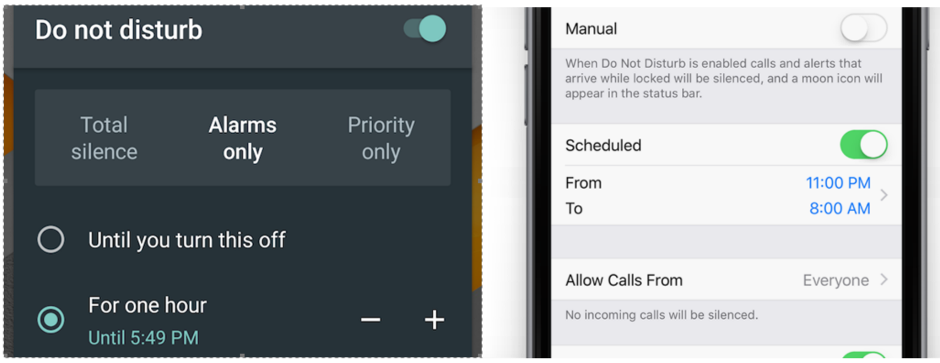

Figure 2.6: "Do not disturb me" in iOS (right) and Android (left)

3 Modeling

In this chapter, the foundations of a new notification brokering mechanism will be explored to find a unique and creative approach to notification based problems. While constructing the design methodology, the most suitable solutions will be hunted after to achieve the main objectives of this work. Before moving onto the modeling process, a clear overview of the problem is beneficiary to ensure the integrity of the research.
After reviewing the problem and the background, the structure of the modeling becomes clearer. As mentioned in the previous chapters, most of the smartphone users have a problem about their notification delivery patterns. Moreover many different statistics and papers support the existence of this issue. As the intention is to tackle this problem computationally, explicit decision steps and their requirements should be taken into account.

3.1 Decision Types and Their Requirements

Before starting to the algorithm design, some basic information should be known by the algorithm to be able to make a decent verdict. As this work approaches to the problem computationally, there is no need to restrict the problem with a single decision type. An advantage of having more than one decisions is that this structure gives the algorithm the flexibility to postpone the decision until a better solution is possible, so there is more room for optimization. In this work, two main decision types are:

- ***Availability decision***
  No matter what the incoming message is, smartphone owners change their focus of attention and in some of the cases this happens only to see that they do not have anything important to pay attention [15]. To avoid such scenarios, inputs for the algorithm should be carefully discovered and marked to find the best moment to send a notification. It should be noted that to find the best moment, availability decision might and probably will require more than one step.

- ***Modality decision***
  After the decision of interruption, a convenient appraisal of the mental workload should be made because previous experiments show that the interruption causes less distraction if the mental busyness is lower [16]. To take advantage of this mechanism, use of the right modalities to convey a notification is necessary and reduces disruption. To decide which modalities to use, a computational model to calculate task interference is necessary.

3.1.1 Availability Decision

To be able to make a well-directed decision, the elements concerning the judgement of the situation should be considered. As the users and their choices vary, some input of the user and their choices should be included in the solution. On the other hand, to optimize communication-disturbance tradeoff the track of number for sent notifications and deleted messages should be kept.

- ***Preferences of the user***
  To decide the degree of the tradeoff between communication and disturbance, the user should be able to decide whether she/he wants a communication oriented, less disturbance oriented or a balanced interaction with incoming messages.

- ***Importance level of applications***
  In a dynamically changing user state and environment, importance level of the applications should be given to the algorithm to be able to decide which message should be marked as disturbance or vice versa.

- ***The number of sent notifications and cache messages***
  Considering an overall state disturbance and communication tradeoff, the track of the number of sent notifications, messages waiting in the queue and messages waiting in the recycling bin should be taken into account to obtain a measurable value for this tradeoff.

3.1.2 Modality Decision

After the decision for availability has been made, there are still a lot of different ways of sending a notification and each different way may result with a different task interference. Hence decision of which modalities to use should be precisely determined. In this case, user state and environmental factors should be included in the decision process. At this stage it is beneficial to remind that after a positive availability decision, the message could still be deleted due to mental workload. More detailed information about the mechanism will be given in the next chapter.

- ***User state***
  Since one of the basic goals of this work is to create a situation aware decision mechanism, the user's current state is a major factor. In this work, considered user states are:

  - Physical state: Walking / Standing / Sitting / Sleeping
  - Visual attention of the user: On screen / Elsewhere
  - Vocal: Speaking / Not speaking
  - Auditory: Listening / Not listening

- ***Environmental state***
  Situation awareness does not only include the state of user and environmental state the user stands is also a prominent factor. In this work, considered environmental states are:
  - Noise level: Loud / Moderate / Silent
  - Illumination level: High / Moderate / Low

3.2 Ways of Sending Notifications

After both type of decision has been made, according to the modality decision, the way of sending the notification will be determined. As any of the readers will be familiar, there are three main ways for a smartphone itself to convey a notification:

- ***Visual***
  Visual notifications are meant to catch human eye's attention and there are several types of visual notifications. Some visual notifications appear as a symbol, some of them as a text and some of them use an internal LED light with different colors.
- ***Auditory***
  This type of notifications could be diversified by using different melodies, volume levels and sound patterns.
- ***Haptic***
  Haptic notifications are widely used to get user attention and they notify user by sending vibrations with different intervals.
- ***Combinations***
  And last but not least, the combination of these three are used as well, depending on the preferences and situation.

3.3 Analogy

Representing the problem by using a catchy analogy is important to provide another point of view, cover the issue widely and simplify the problem to a familiar case, which is a web server. Web servers operate to answer many different requests with a limited memory and the workload of this requests could easily exceed the capacity of the server [17]. A human being's limited cognitive capacity and attention demanding notifications from numerous applications build a similar scenario from this perspective.
Basically a web browser sends a request to the web server via HTTP and asks for a specific resource present in that server. Web server could respond to this request with two different options. The first option is that the server could respond with the requested resource or document, while in the second options server is not available to respond and an error message to the user is sent. A web server provides service

to many different customers at the same time [18] which may result in delay and congestion due to different factors such as queues in routers, limited capacity of server, connection problems, etc.
Looking from this perspective, notification brokering problems are quite similar to web server based issues where the web server represents a human being and mobile applications represent different users sending requests to web server. Different mobile applications are sending notifications and expect the user to attend somehow and moreover human cognitive system have a limited capacity like a server. This situation is demonstrated by Figure 3.1.

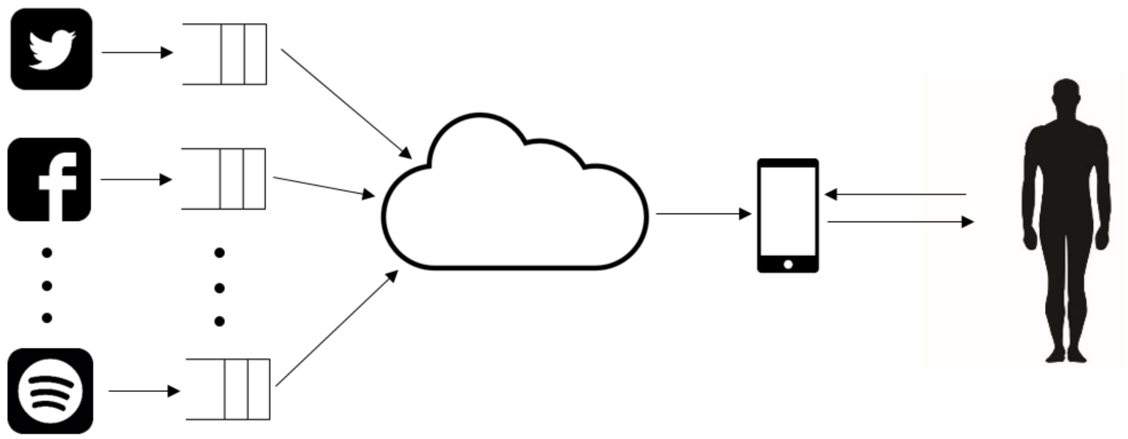

Figure 3.1: Representation of Analogy

To solve this problem with a similar approach, some missing elements should be added to the notification brokering system from web server structure. For example when the demand is very intense, usually some queues at the routers occur. So in similar cases such as many applications bombarding the user with notifications, they might classified and be placed in a queueing system just like routers do, in order to maximize the communication and user's responsiveness. When a web browser receives an error message, this means that server is either too busy or the requested resource is not available at the server. To complete the big picture, this case could be replaced with “the user does not have enough cognitive response capacity, do not send any notifications".

4 System Design

In this chapter, system elements and the system itself as a notification broker will be explained and examined. Firstly the input data and its properties will be detailed, after that designed notification brokering algorithm will be shown and its components will be investigated.

4.1 Input Data

- ***User and Environment State***
  Since the aim is to make the optimization according to the user state, there is a need to have valuable information about what user is doing and in what kind of environment the user is present. System ordinarily checks and updates the user and environmental state when any sort of change happens in the following terms:
  - Physical state of user
  - Visual attention of user
  - Vocal state of user
  - Auditory state of user
  - Noise level of environment
  - Illumination level of environment
- ***Incoming Messages***
  Incoming messages are at the core of the information awareness because they are the actual things to be scheduled on-line. When a message arrives to the smartphone, the following information are passed to the algorithm:
  - Arrival time for the new message
  - Application name
  - Required viewing time for the notification

Note: It is obvious that there are a lot of different influencing factors such as other physical measurements, location, connection type, signal strength, social factors etc. [6] but because of the scope of this thesis and for the sake of simplicity, the number of input data types are limited to a certain amount.

4.2 Notification Brokering Algorithm and Its Components

The notification brokering system consists of different components and each component has its own importance in terms of optimization. Since the decision in on-line scheduling problems are irreversible, having more than one steps for user's availability decision makes this system more resilience for dynamic changes and prevents unwanted decisions. Additionally a decision made with on-line scheduling algorithm will

be no better than an off-line algorithm decision and the best performance evaluation is made by assuming a specific probabilistic distribution for discrete cases [19], such as the probability to be viewed for a notification, constants for different priorities or queueing and recycling mechanism.
First of all, incoming messages come to the first decision step, a.k.a. queue decision, and this step decides whether the message is qualified to continue to the queueing system or not. The decision is made according to two different score types and user's preference of communication/disturbance ratio. After a positive queue decision, the processed message is placed into a queue according to its priority. Messages leave their queue according to the queue priority, so if there are any messages present in the priority 1 queue, none of the messages in priority 2 or priority 3 queue could proceed to the next step.
After a message leaves the queue, interference calculation is made by using multiple resource theory model. If the calculated total interference between the incoming message's resource requirements and the user's current state is at an acceptable level, the modality decision will be made in order to determine how the notification should be conveyed and the message is sent to the user as a notification. If the interference level is more than user could handle, the message is sent to the recycling bin, where messages are recycled with a probabilistic structure, according to their priorities in order to increase the message circulation in the system and break the irreversible decision rule in on-line scheduling mechanisms. Figure 4.1 represents the notification brokering algorithm and its working principle.

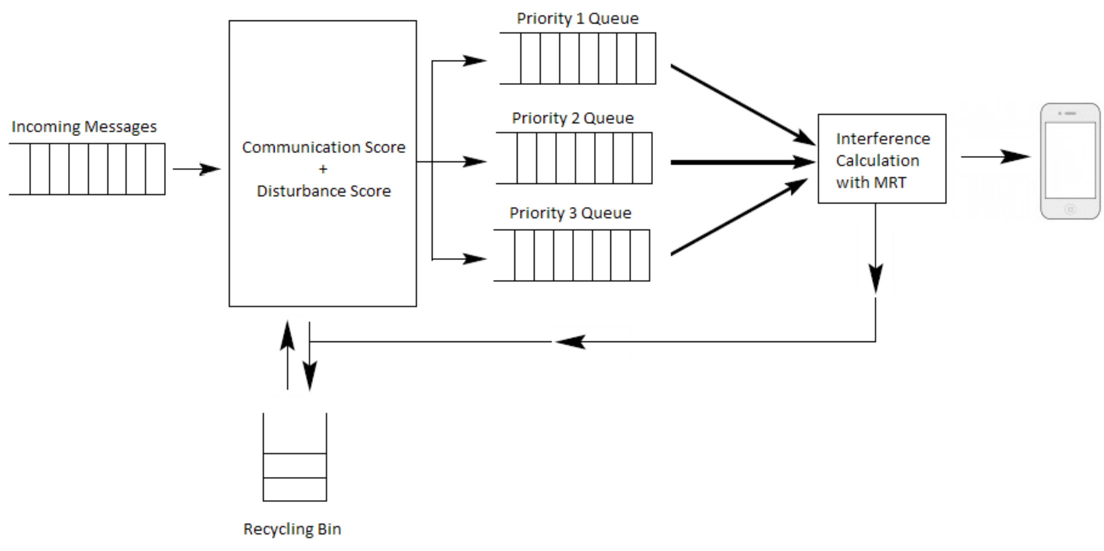

Figure 4.1: Block Diagram of the Notification Brokering Algorithm

4.2.1 Queue Decision

When a new message arrives into the system, it should be assessed according to the existing conditions to make the decision about the message's future. If the decision is positive, that means this message is eligible to be placed into the queueing system, otherwise the message will be sent to the recycling bin. While making this decision, several ingredients should be calculated and investigated.

User Preference Ratio

As the performance of the notification algorithm is highly associated with assisting attention designation [w], the preference of the user plays a great role in decision making. User preference ratio, as we can understand from its name, simply defines where the user stands on a 1-D axis which points out the tradeoff between information awareness/communication and mental workload/disturbance. If the user moves on the axis to the left side, it means that the user wants to be informed as much as possible and moving to the right means she/he does not want to be disturbed if it is not extremely necessary. As it could be predicted easily, a preference on the middle of the axis balances the tradeoff equally. Figure 4.2 demonstrates how a user could select her/his preference.

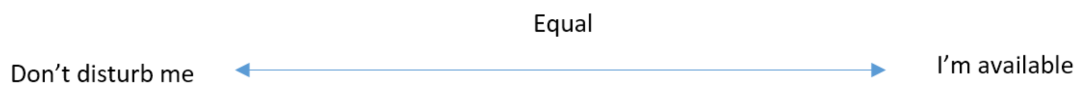

Figure 4.2: Representation of User Preference on 1-D Axis

After the selection is made, user preference ratio is calculated according to the following pseudo-code:

IF UserSelection == 0
    $\beta = 1$
    $\alpha = 1$
ELSEIF UserSelection > 0
    $\beta = 1$
    $\alpha = 1 +$ UserSelection
ELSE
    $\beta =$ -1 + UserSelection
    $\alpha = 1$
ENDIF
UserPreferenceRatio = abs(α / β)

Communication Score
Communication score keeps track of the messages present in the system and gives an overall idea about the degree of information awareness, because it is believed that arranges communication actions are capable of decreasing the mental workload of the user [20]. This score is normalized to [0 1] interval to make the comparison and to have standard boundaries. The smaller this score gets, the higher communication degree is. Communication score is calculated by going through each message in the system. If the message is sent to a user, the score gets smaller with a value proportioned to the involved message's priority value. On the other hand if the message is sent to recycling bin the score gets bigger with the same value. A message waiting in the queue harms the score as half as a message in recycling bin since waiting messages are delaying the communication and lowering the quality of the communication. Communication score is calculated according to the formula given below.

$$Score_{communication} = \frac{-\sum_{i=1}^{u} b * Priority_i + \sum_{j=1}^{q} c * Priority_j + \sum_{k=1}^{r} d * Priority_k}{u+q+r} \tag{1}$$

Where u is the number of messages sent to user, q is the number of messages in the queue and r is the number of messages sent to recycling bin. Priority constants might change according to how big the difference between different priorities are, but they are assumed to be 10, 8 and 6 for priority 1, priority 2 and priority 3, respectively. The values of a, b and c are used to change the effects of queue, sent or recycle messages in the whole system and as default, values of b, c and d are assumed 1, 0.5 and 1 respectively.

Disturbance Score
Disturbance score indicates how much the user is disturbed overall by using the same method with communication score. Again, the score is normalized between 0 and 1, and if the score is getting smaller that means the user is being disturbed less. The calculation of this score is almost same as communication score. The only difference is sent messages are magnifying and deleted messages are diminishing the score. Messages waiting in the queue has the same effect on this score as well since if a messages is waiting in the queue, that message could be counted as a potential disturbance in the future. Disturbance score is calculated according to the formula below.

$$Score_{communication} = \frac{\sum_{i=1}^{u} b * Priority_i + \sum_{j=1}^{q} c * Priority_j - \sum_{k=1}^{r} d * Priority_k}{u+q+r} \tag{2}$$

Where u is the number of messages sent to user, q is the number of messages in the queue and r is the number of messages sent to recycling bin. Priority constants might change according to how big the difference between different priorities are, but they are assumed to be 10, 8 and 6 for priority 1, priority 2 and priority 3, respectively. The values of a, b and c are used to change the effects of queue, sent or recycle messages in the whole system and as default, values of b, c and d are assumed 1, 0.5 and 1 respectively.

Decision Criteria

To make the initial decision for queueing, the algorithm compares the user preference ratio to the ratio of the scores which is communication score divided by disturbance score. This simple relation roughly gives an idea about how much user preference is matching with the actual situation in the system. According to these ratios, some general statements about the existing condition could be made and these ratios are expressed below.

$$\frac{\alpha}{\beta} \approx \frac{Score_{communication}}{Score_{disturbance}} \tag{3}$$

- Case U1: User ratio $> 1 \longrightarrow$ User doesn't want to be disturbed
- Case U2: User ratio $< 1 \longrightarrow$ User wants more communication and is okay with some disturbance
- Case S1: Score ratio $< 1 \longrightarrow$ User is receiving relatively more messages and disturbed to some extent
- Case S2: Score ratio $> 1 \longrightarrow$ User is not being disturbed much

If cases U1 and S1 happen together, that means there is a mismatch in between user preferences and system situation and this needs to be fixed. In this case user doesn't want to be disturbed but she/he is receiving relatively more notifications. To fix the situation and change the trend in a desired way, the message should be sent to recycling bin.

On the other hand if cases U2 and S2 occur at the same time, it is an indicator of another mismatch situation in which user wants to receive more notification but she/he is not being disturbed much. In this case the message will be forwarded to the queueing system. Until now two different mismatch scenarios are examined and in such cases making a decision is relatively easier but in most of the realistic scenarios this sort of mismatch scenarios are occurring less frequently. Since the four cases stated above are quite general statements, mismatch scenarios might be thought as a "safety belt".

If the case combination is building up a matching scenario, which are U1 & S2 and U2 & S1, another approach should be used to make a decent decision. Obviously there is a need to mark the situation of the system more precisely. Since the absolute value of α reflects the user's degree preference of "don't disturb me" and the absolute value of β reflects the user's degree of "you can disturb me", finding the location of

score ratio (which is the situation of the system defined by previous decisions) on the user's preference space will make the decision much easier.
In the matching scenario, the decision space could be divided into three main parts: send to recycling bin, send to queue, apply the same as previous decision. To apply the previous decision again for another newly arrived message, the score ratio should not be deviated much from the user ratio. This deviation might be changed according to the desired flexibility. In this work this value is considered as 0.2*user ratio, which is derived by empirical trials and represented by λ. If the score is deviated more than $-\lambda$ the message will be sent to recycling bin and if the score is deviated more than λ the message will be included in the queue. Figure 4.3 represents the matching scenario and its consequences.

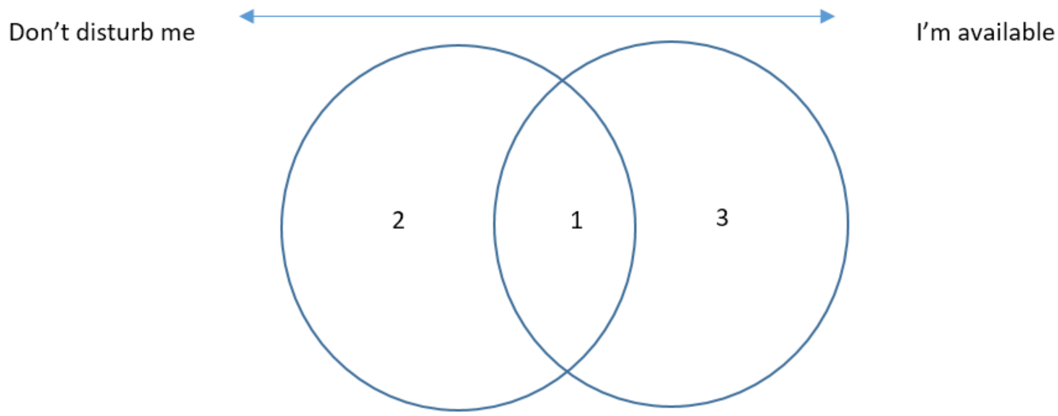

Figure 4.3: Decision Space for Matching Scenarios

- 1 $\longrightarrow$ User Ratio $- \lambda <$ Score Ratio $<$ User ratio $+ \lambda$ $\longrightarrow$ Apply the same decision as previous
- 2 $\longrightarrow$ Score Ratio $<$ User Ratio $- \lambda$ $\longrightarrow$ Send to recycling bin
- 3 $\longrightarrow$ Score Ratio $>$ User Ratio $+ \lambda$ $\longrightarrow$ Send to queue

4.2.2 Queueing and Recycling System

Queue and recycling system are important in terms of transferring the data efficiently. In this part, properties of queues and recycling bin will be examined.

Queueing Mechanism
Queueing mechanism consists of three different queues working with FIFO (First In First Out) mechanism, where messages leave the queue with the same order as they enter, to consider the incoming time and reduce the delay. We know that small delays in notification systems do not harm the user since even important notifications do not ask for prompt care and the correlation between being important and requiring urgent care is poor [11]. After a message fulfills the decision criteria, it needs to be placed in a queue. Since messages are classified according to their priorities (1, 2 or 3), messages with different priorities are held in different queues. Afterwards they are transmitted to the next stage to have their dual task interference calculated. The transmission of messages to the next stage is done with a priority scheduling logic since notification scheduling could be improved by considering the importance of the information [21]. In communication based problems, a bottleneck is happening at most of the times and user is also receiving unnecessary messages. As we use different queues for different types of notifications, the order of feeding mechanism is controlled by priority of notifications. For example if any messages are waiting in the first queue, other queues are suspended until the first queue is empty. If another queueing mechanism or a single queue was used in such structure, the system will end up spending the same resources for all of the notifications which is not efficient at all. Figure 4.4 represents the priority and working mechanism of queues.

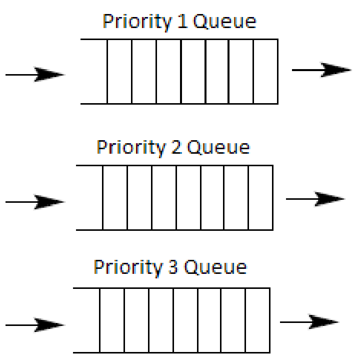

Figure 4.4: Queue Structure for Notifications with Different Priorities

Recycling Mechanism
First of all, recycling stack is a place to accumulate the messages which are not able to fulfill the criteria to reach the user, such as mental workload or low priority some of the messages get the chance to reach to user if necessary conditions are present. The method for recycling will be detailed in the upcoming part.
The most important difference between recycling structure and priority queues is

that recycling works with LIFO (Last In First Out) mechanism, which is also known as “stack”. In this way most recently elements in the recycling bin are faster to reach and ready to be recycled while the scheduling process is continuing. Figure 4.5 demonstrates the structure of a stack.

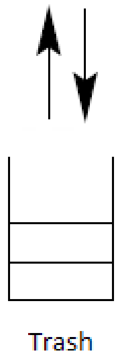

Figure 4.5: Recycling Bin and Its Working Mechanism

After understanding the working principle of recycling stack, building a recycling mechanism will be easier. At this point it is beneficial to remind that a stack structure is very efficient in terms of reaching the last element without using any pointers but at the same time it makes harder to recycle the messages at the bottom. So in the design process, this tradeoff is addressed by using a probabilistic and priority based structure again, in order to not to miss any important messages.
Recycling is necessary in terms of making the most of incoming messages since some inappropriate decisions could occur sometimes in such probabilistic systems. The same priority logic is also valid here so messages with higher priorities could get a higher chance to be recycled comparing to low priority messages. As mentioned in the previous sentence, recycling is done with a probabilistic approach and in each recycling iteration, the maximum number of one message is recycled since the algorithm itself works in a loop. Basically each time a new message arrives, a message or no messages will be recycled into the system. The mechanism of recycling is given below:

- Go through the recycling stack and look for priority 1 messages. If a priority 1 message is found recycle the message and finish the iteration.

- If there is no priority message is found, go through the recycling stack with 50% probability and recycle a priority 2 message if found.

- Else go through the recycling stack with 20% probability and recycle if a priority 3 message if found.

4.2.3 Interference Calculation and Modality Decision

In this part, the principle, used methods and techniques of interference calculation and modality decision will be viewed. To make the modality decision, a numerical value of the interference should be known and according to this numerical value modality decision will be made.

Interference Calculation
To obtain a decent interference calculation, multiple resource theory is used since this model is able to reckon the interference quantitatively. The computation is made through the cube presented in the upcoming figure, starting from the left side and it includes three stages which are perception, cognition and responding.
Before explaining these stages, it is important to point out that dividing the attention between different modalities is more effective than dividing the attention in the same modality. For example it is easier the focus on two tasks where one of them requires auditory attention and the other one requires visual attention and this phenomena is called cross-modal time sharing. If the user tries to divide the attention between two different auditory channels, which is intra-modal time sharing, the success ratio will be less [8]. Additionally there are two different visual channels (focal and ambient) to cover the complex structure of visual perception.

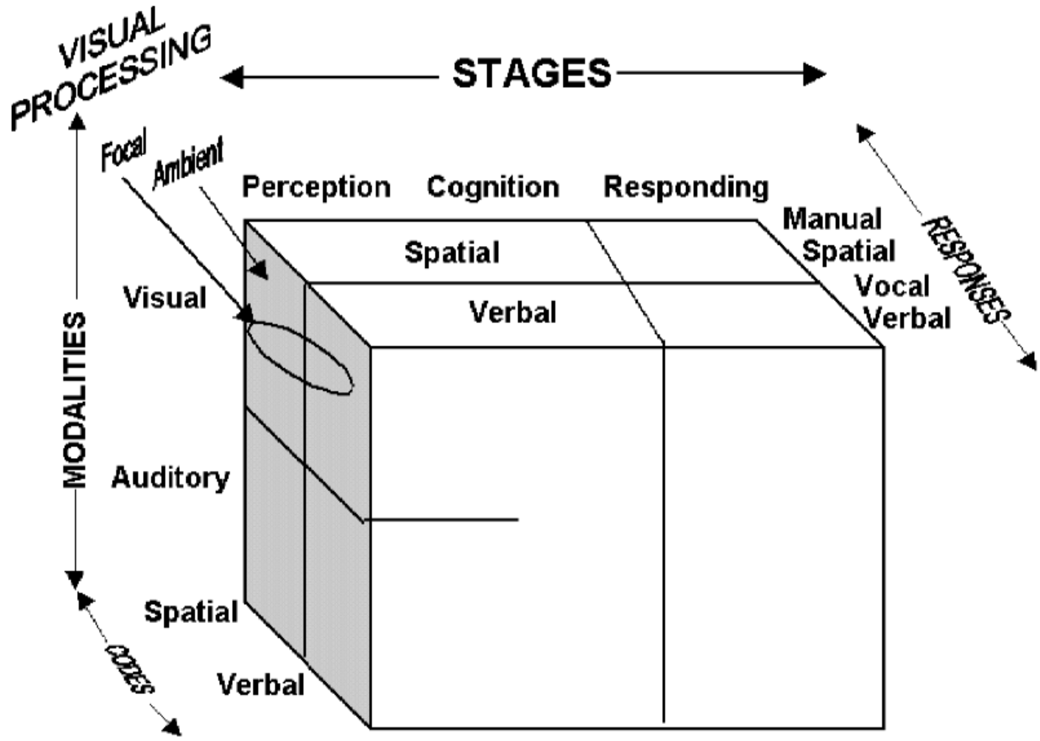

Figure 4.6: 3-D Representation of Multiple Resource Theory [8]

In the perception stage, there are two different codes and two different modalities available and the combination of these will define the type of time sharing tasks. At the first sight, perception step might look a bit ambiguous but thanks to P.A. Hancock et. al. it becomes easier to understand the similarities and differences with 4 realistic examples and Figure 4.7 demonstrates this idea [22].

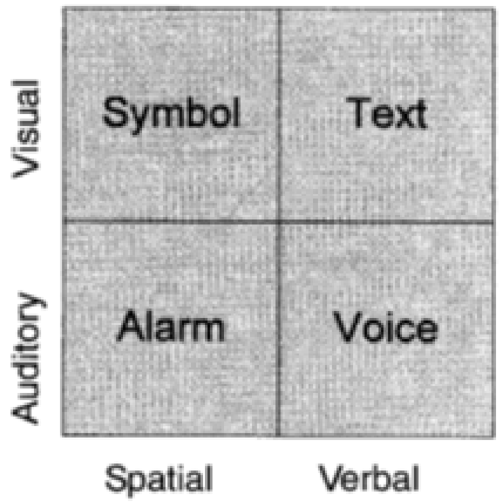

Figure 4.7: Example Outcomes of Modality and Code Combinations [22]

After being able to label the scenario in terms of perception type, there is a need for a computational process to determine the interference quantitatively. The following steps represent how the computational model works.

⟶***Step 1: Total demand calculation***

Before moving to the computation of the dual task interference, the demand of related jobs should be calculated as separate vectors. The same variables in the conflict model is used in this case with a small difference. The marking of the task values are made from 0 to 3, where 0 means there is no demand, 1 means some demand ,2 means significant demand and 3 means extreme demand. Empty spaces in the chart should be filled and the mean demand for task 1 and task 2 should be calculated. To be able to determine the right values for processing stages, a task analysis should be performed beforehand.

Table 4.1: Demand Calculation for Tasks

	V_f	V_a	A_s	A_v	C_s	C_v	R_s	R_v	Mean
Task 1									
Task 2									

where
V=Visual, A=Auditory, C=Cognitive, R=Response, f=Focal, a=Ambient, s=Spatial, v=Verbal

$$Mean(Task_x) = (V_f + V_a + A_s + A_v + C_s + C_v + R_s + R_v)/8 \tag{4}$$

$$TotalDemand = Mean(Task_1) + Mean(Task_2) \tag{5}$$

⟶***Step 2: Total conflict calculation***

Second step is to calculate the conflict between two tasks. To succeed the first step of computational multiple resource theory model, the conflict matrix should be used to determine which tasks may overlap in terms of mental workload through the three stages defined earlier. The values of the matrix given below are constructed by Horrey [8] and used in a study by Horrey and Wickens to calculate the task interference and in-vehicle performance [23]. The cells with a value of more than zero are marked in the conflict matrix and the non-zero intersections are added together.

Table 4.2: Conflict Matrix for Multiple Resource Theory [8]

	V_f	V_a	A_s	A_v	C_s	C_v	R_s	R_v
V_f	0.8	0.6	0.6	0.4	0.7	0.5	0.4	0.2
V_a	X	0.8	0.4	0.6	0.5	0.7	0.2	0.4
A_s	X	X	0.8	0.4	0.7	0.5	0.4	0.2
A_v	X	X	X	0.8	0.5	0.7	0.2	0.4
C_s	X	X	X	X	0.8	0.6	0.6	0.4
C_v	X	X	X	X	X	0.8	0.4	0.6
R_s	X	X	X	X	X	X	0.8	0.6
R_v	X	X	X	X	X	X	X	1

Sum of all the cells in matrix is 20, hence a normalization is necessary in order to balance the effect of the demand and conflict. Since the maximum value for demand is 6, multiplying the total conflict with 0.3 will normalize the result between 0 and 6.

⟶***Step 3: Total interference calculation***

This is the easiest step of and formula below gives the final outcome for the computational model of the multiple resource theory. Since values are normalized, they affect the total interference with the same rate.

$$TotalInterference = TotalDemand + TotalConflict \tag{6}$$

An Example of Dual-Task Interference

To represent the problem clearly, an example dual task interference will be calculated. Let's assume that the user is sitting in a coffee shop while speaking to the phone using a Bluetooth headset.

In this case user status will be: Sitting, speaking and visual focus on environment. The environmental status could be assumed as: Medium illumination and noisy. Values for Task 1 will be determined according to the user and environment status. Each modality combination requires a separate calculation, so let's only calculate the interference if a notification is sent by visual and auditory modalities.

Table 4.3: Example Demand Calculation

	V_f	V_a	A_s	A_v	C_s	C_v	R_s	R_v	Mean
Task 1	0	1	2	0	2	1	1	2	
Task 2	1	0	1	0	1	1	0	0	

$$Mean(Task_1) = (1 + 2 + 2 + 1 + 1 + 2)/8 = 9/8 = 1.1250 \tag{7}$$

$$Mean(Task_2) = (1 + 1 + 1 + 1)/8 = 4/8 = 0.5 \tag{8}$$

$$TotalDemand = 1.1250 + 0.5 = 1.6250 \tag{9}$$

Table 4.4: Example Conflict Matrix for Multiple Resource Theory

	V_f	V_a	A_s	A_v	C_s	C_v	R_s	R_v
V_f	0.8	0.6	0.6	0.4	0.7	0.5	0.4	0.2
V_a	X	0.8	0.4	0.6	0.5	0.7	0.2	0.4
A_s	X	X	0.8	0.4	0.7	0.5	0.4	0.2
A_v	X	X	X	0.8	0.5	0.7	0.2	0.4
C_s	X	X	X	X	0.8	0.6	0.6	0.4
C_v	X	X	X	X	X	0.8	0.4	0.6
R_s	X	X	X	X	X	X	0.8	0.6
R_v	X	X	X	X	X	X	X	1

$$\begin{aligned} TotalConflict = (&0.6 + 0.6 + 0.7 + 0.5 + 0.4 + 0.2 + 0.8 + 0.7 + 0.5 + 0.4 + 0.2 + 0.8 \\ &+ 0.6 + 0.6 + 0.4 + 0.8 + 0.4 + 0.6) * 0.3 = 2.94 \end{aligned} \tag{10}$$

$$TotalInterference = TotalDemand + TotalConflict = 1.6250 + 2.94 = 4.56 \tag{11}$$

Modality Decision

After a message fulfills all the necessary requirements and passes the required stages, the challenge is to make the last decision and this judgement is the gateway to the user. It has been shown with experiments that the error rate in the main task has no significant correlation with the notification modality. On the other hand in the same study the results point out that the modality of the notification will affect the proportion of the activity [24]. So it is appropriate to say that the modality decision affects the user's working performance and attention. Since we both have the user's disturbance preference and a numerical value for dual task interference, the modality decision could be made on the mark and there is no problem with disturbing the user accordingly to her/his preference (User Preference Ratio).

To make the modality decision, the following steps are followed: Interference value for different modality combinations are calculated for 7 different combinations:

Interference Array = $[I_a\ I_v\ I_h\ I_{av}\ I_{ah}\ I_{vh}\ I_{avh}]$

Interference array is sorted. The corresponding interval user preference of is found on the sorted interference array.

Example:
UserSelection = 2.6
SortedInterferenceArray = $[I_h\ I_v\ I_a\ I_{vh}\ I_{ah}\ I_{av}\ I_{avh}]$
UserSelection = UserSelection + 4 (to normalize between 0 and 8)
Interval = floor (level * 7/8 + 1) = 5
InterferenceValue = SortedInterferenceArray(Interval) = I_{ah}
SelectedModality = Auditory & Haptic

Additionally there is no obligation of sending the notification if the total interference is much more than user can handle. In this case the message will be sent to recycling stack and still have the chance to be recycled if the total interference is greater than the threshold value set.

To decide whether to send the notification to the user or to the recycling bin, another priority based approach is used. To be able to make this decision, the maximum and minimum interference a notification could create should be known. According to the computational model, the interference value will be between 0 and 12. On the other hand we also know that bigger values of interference is quite extreme so the interference value of all the possible notification modalities and user states should be calculated in order to know the maximum and minimum interference values.

- If NotificationPriority==1 && Interference $<$ x $\longrightarrow$ send the notification
- If NotificationPriority ==2 && Interference $<$ y $\longrightarrow$ send the notification
- If NotificationPriority ==3 && Interference $<$ z $\longrightarrow$ send the notification

Where x $>$ y $>$ z, which means notifications with a higher probability could be sent to the user with bigger interference values.

4.2.4 Viewing Notifications

Since one of the main objectives of the thesis is to provide more communication with less effort, viewing of the messages is an important factor to increase the information awareness of the user. If the user gets interrupted with another message while attending to a notification, it definitely will affect the experience in a bad way.
Also it has been pointed out in the previous studies [1], [11], [6] the time between interruptions affect the task success rate significantly, and thus there is a need for a probabilistic decision feature for to decide whether the user is going to view the notification or not. Moreover it has been assumed that if the user is already using the device, it is very likely that user will see the notification and react in the first 30 seconds to 50% of the notifications [b] so there is also a need to estimate the interaction time to view a message in different cases.

Viewing Probability
To combine the scientific information we have in a decision mechanism, two main influencing factors for the viewing should be considered which are the importance level of the notification and the visual attention focus of the user. The visual focus of attention and the importance of the notification are always known so the probability of viewing to the notification is calculated according to the following statements:

- If NotificationPriority==1 && VisualAttention == Screen $\longrightarrow P_{viewing}$=0.9
- If NotificationPriority ==1 && VisualAttention == Environment $\longrightarrow P_{viewing}$=0.6
- If NotificationPriority ==2 && VisualAttention == Screen $\longrightarrow P_{viewing}$=0.7
- If NotificationPriority ==2 && VisualAttention == Environment $\longrightarrow P_{viewing}$=0.4
- If NotificationPriority ==3 && VisualAttention == Screen $\longrightarrow P_{viewing}$=0.5
- If NotificationPriority ==3 && VisualAttention == Environment $\longrightarrow P_{viewing}$=0.2

Viewing Time
After having the necessary probabilistic structure, there is a need to estimate the interaction time to view a message. As we know that more than 70% of the notifications are first viewed through the notification drawer [12], the estimation time could be interpreted by using keystroke level model (KLM) and the two separate cases could either be the screen is on or the screen is off. If user's visual attention is on the screen, it has been assumed that the screen is on and if user's visual attention is elsewhere, it has been assumed that the screen is off.
The keystroke level model is used to estimate the time for a user to complete a task related to computer systems, by considering keystrokes, gestures and mental preparations [25].

– KLM sequence for off-screen case

  1. Mental perception (M) = 1.2 sec.

2. Homing to the smartphone (H) = 0.4 sec.
3. Pointing the finger to notification drawer (P) = 0.7 sec.
4. Slide the drawer (S) = 0.5 sec.
5. Reading (R) = 3 sec.

$$T_{viewing} = T_M + T_H + T_P + T_S + T_R = 1.2 + 0.4 + 0.7 + 0.5 + 3 = 5.8sec. \quad (12)$$

– KLM sequence for on-screen case: Since the user can always choose the way of showing notification when the screen is on, we assume that the notification is shown on a banner automatically.

1. Mental perception (M) = 1.2 sec.
2. Reading (R) = 3 sec.

$$T_{viewing} = T_M + T_R = 1.2 + 3 = 4.2sec. \quad (13)$$

According to the obtained information; if the user is attending to the message, she/he will not be disturbed with another notification until $T_{viewing}$ is completed, since the queueing system is capable to handle the traffic well.

4.2.5 Graphical User Interface

In complex decision systems, following the results and presenting a demo could be complicated sometimes. To avoid such scenario and to provide an understandable work, a graphical user interface is also designed and implemented.
The interface consists of five main parts and related properties are grouped together. Before starting the notification system, user can select her/his preference of disturbance ratio and press start to launch the algorithm. The section on the left top corner informs the user about the current time, user status and environmental status such as visual focus, speaking ad physical status of user and illumination and noise of the environment. Two score types are grouped together in the scores section so communication score and disturbance score could be followed during the process. Additionally it is possible to see how many messages are waiting in queues or recycling bin and how many messages are sent as notifications. Lastly in the notification section, the name and the priority of the application showed and used modalities for the notifications are marked with green.

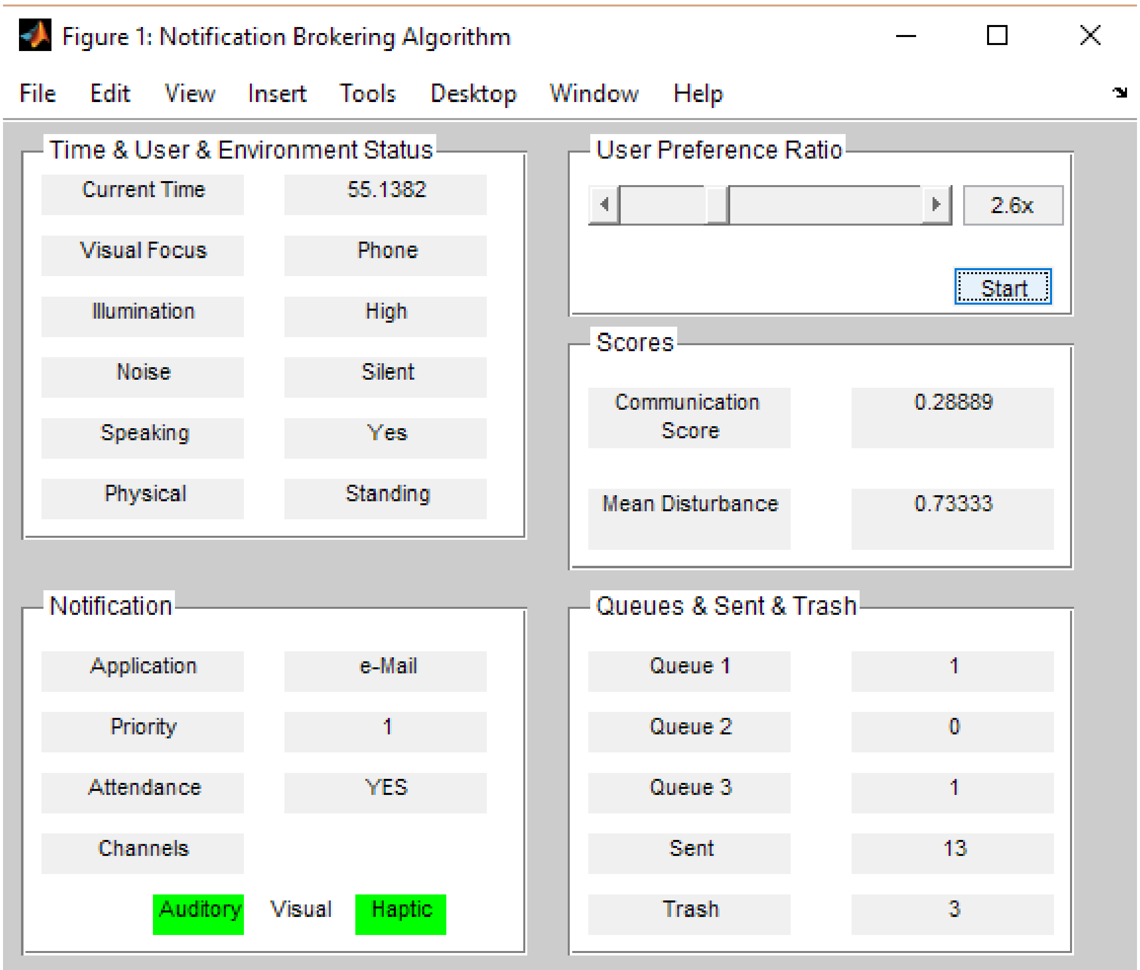

Figure 4.8: Graphical User Interface

5 Results

In this part, simulations for different scenarios will be run and the results will be given. Each user scenario will be run with different incoming message intervals in order to see how the algorithm reacts in different situations.
Note: Maximum theoretical value of interference is 12, but that is an extreme case. To be able to judge the interference value fairly, all of the possible scenario combinations and their interference values are calculated by writing a MATLAB script. For this chapter, the possible interference with a notification and a task is calculated as 6.

5.1 Scenario 1

Antti is working on his desk on standing position, his visual focus changes between the computer screen and the smartphone. His room is silent and he doesn't want to be disturbed except important things. User preference ratio = 2.6 $\longrightarrow \alpha = 2.6$, $\beta = 1$

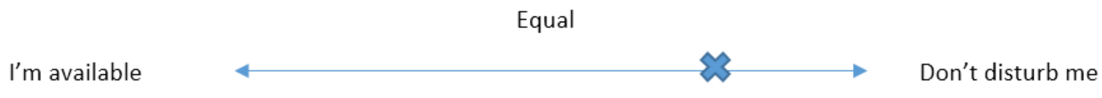

Figure 5.1: Representation of User Preference on 1-D Axis

Table 5.1: User States for Scenario 1

Time	Visual Focus	Illumination	Noise	Speaking	Physical
0 – 60 sec.	Elsewhere	Mid	Silent	No	Standing
60 – 120 sec.	Phone	Mid	Silent	No	Standing

5.1.1 Interval 1

A new message arrives with a mean of 2.5 seconds

Table 5.2: Incoming Messages and Their Allocations

	# messages	# notifications	# recycling bin	# queue
Priority 1	7	6	1	0
Priority 2	22	9	12	1
Priority 3	17	0	17	0

- Mean delay for a notification = 32.1 sec.
- Standard deviation for delay = 37.35 sec.
- Mean disturbance for a notification= 1.83
- Used modalities: Haptic (100%)

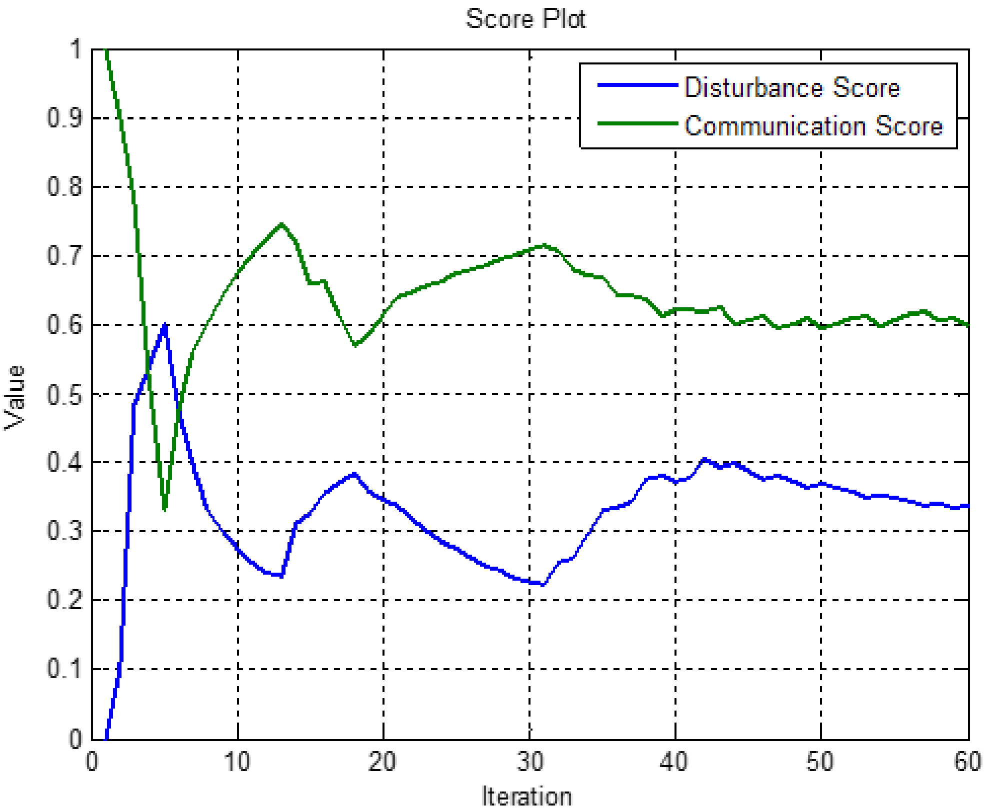

Figure 5.2: Communication and Disturbance Score Development

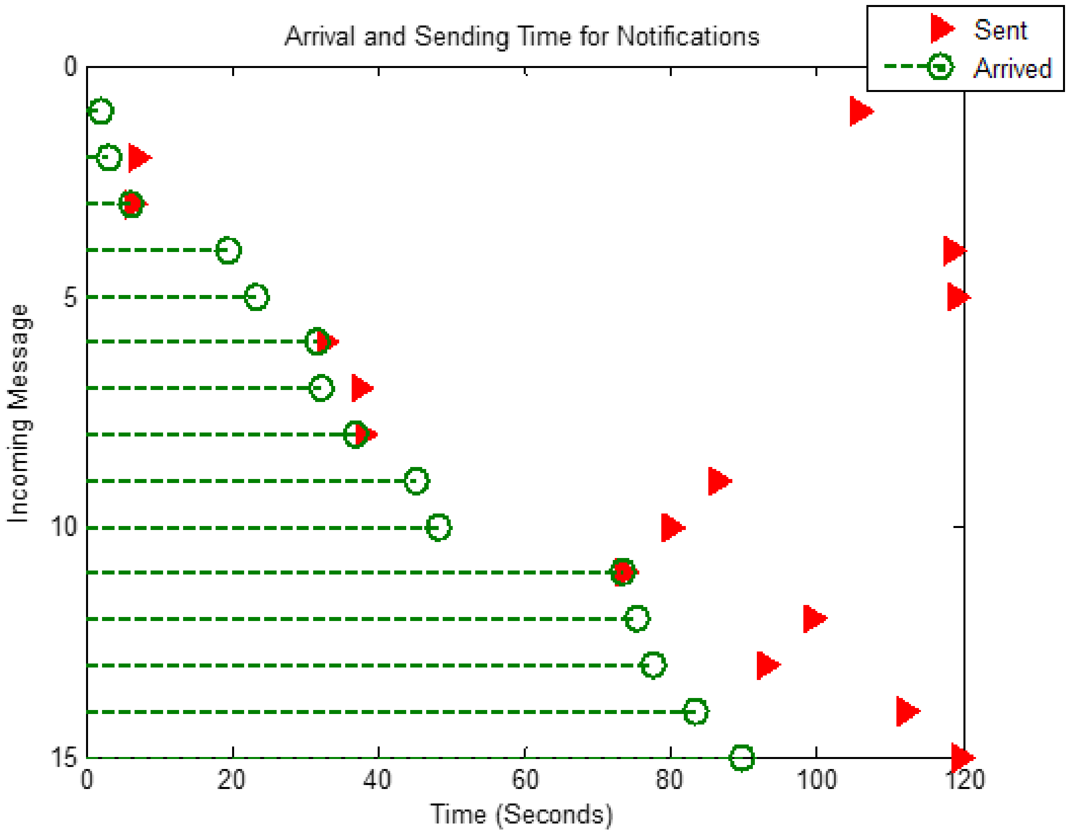

Figure 5.3: Arrival and Sending Time of Notifications

5.1.2 Interval 2

A new message arrives with a mean of 5 seconds

Table 5.3: Incoming Messages and Their Allocations

	# messages	# notifications	# recycling bin	# queue
Priority 1	7	4	3	0
Priority 2	10	2	8	0
Priority 3	7	1	6	0

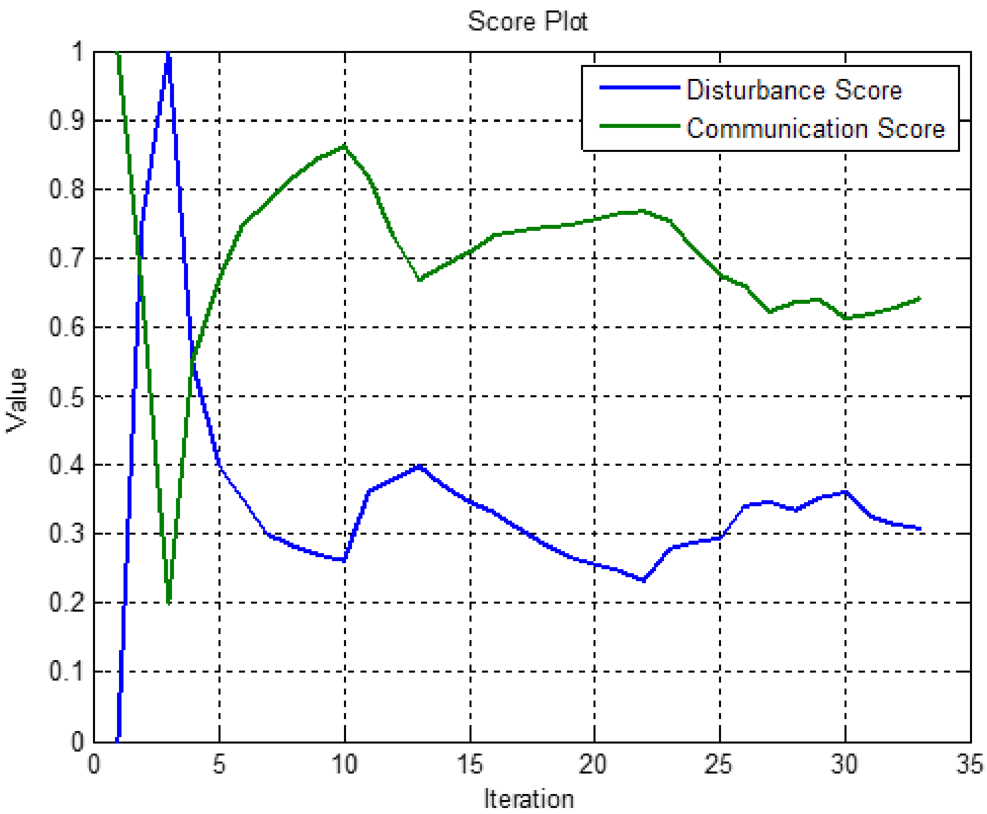

Figure 5.4: Communication and Disturbance Score Development

- Mean delay for a notification = 19.18 sec.
- Standard deviation for delay = 31.65 sec.
- Mean disturbance for a notification= 1.78
- Used modalities: Haptic (100%)

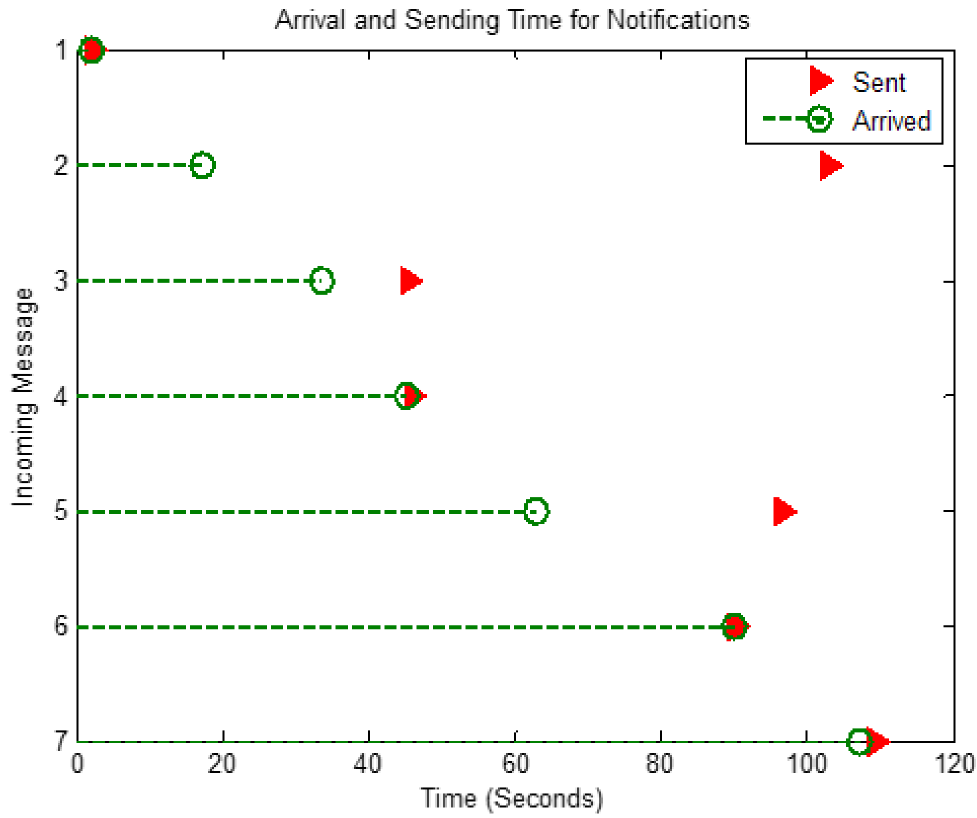

Figure 5.5: Arrival and Sending Time of Notifications

5.2 Scenario 2

Anna is sitting in a coffee shop and sometimes speaks on the phone using her Bluetooth headset. She wants the communication and disturbance to be equal. User preference ratio $= 1 \longrightarrow \alpha = 1, \beta = 1$

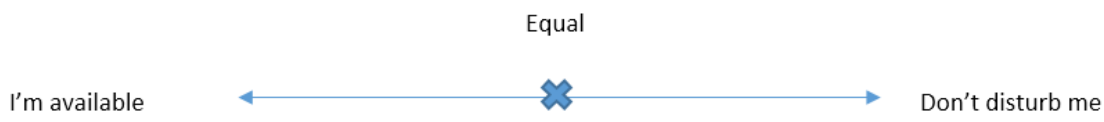

Figure 5.6: Representation of User Preference on 1-D Axis

Table 5.4: User States for Scenario 2

Time	Visual Focus	Illumination	Noise	Speaking	Physical
0 – 60 sec.	Elsewhere	Mid	Mid	Yes	Sitting
60 – 120 sec.	Phone	Mid	Mid	No	Sitting

5.2.1 Interval 1

A new message arrives with a mean of 2.5 seconds

Table 5.5: Incoming Messages and Their Allocations

	# messages	# notifications	# recycling bin	# queue
Priority 1	7	7	0	0
Priority 2	22	13	8	1
Priority 3	15	0	14	1

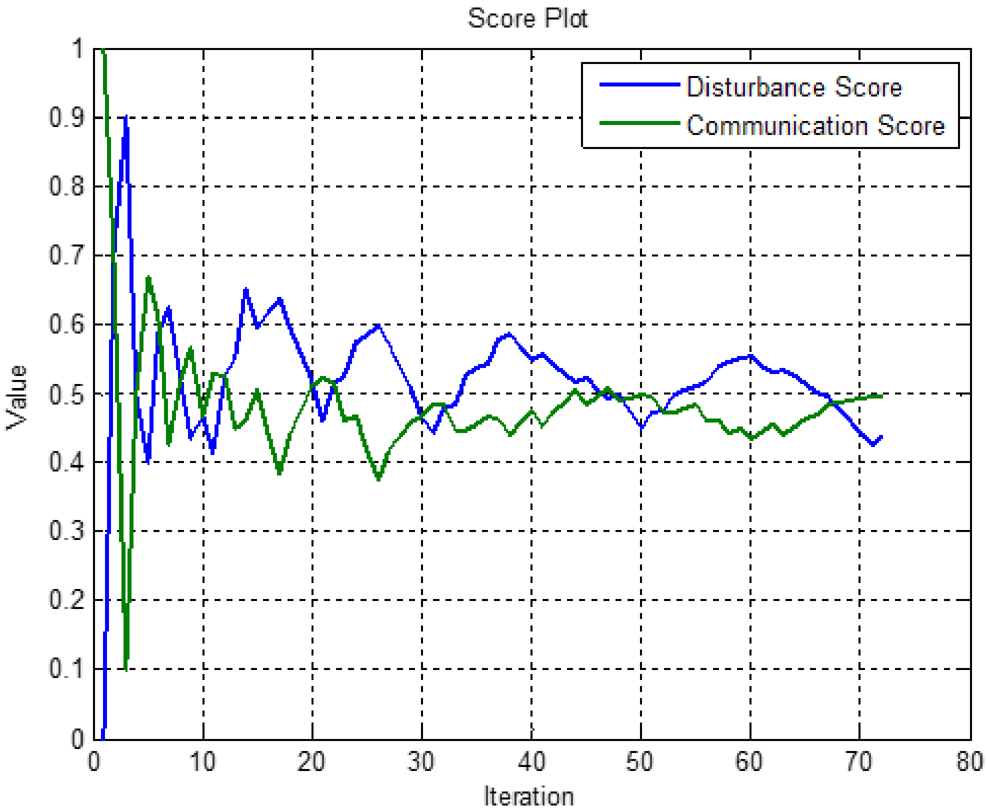

Figure 5.7: Communication and Disturbance Score Development

- Mean delay for a notification = 17.86 sec.
- Standard deviation for delay = 22.65 sec.
- Mean disturbance for a notification= 2.81
- Used modalities: Visual + Haptic (100%)

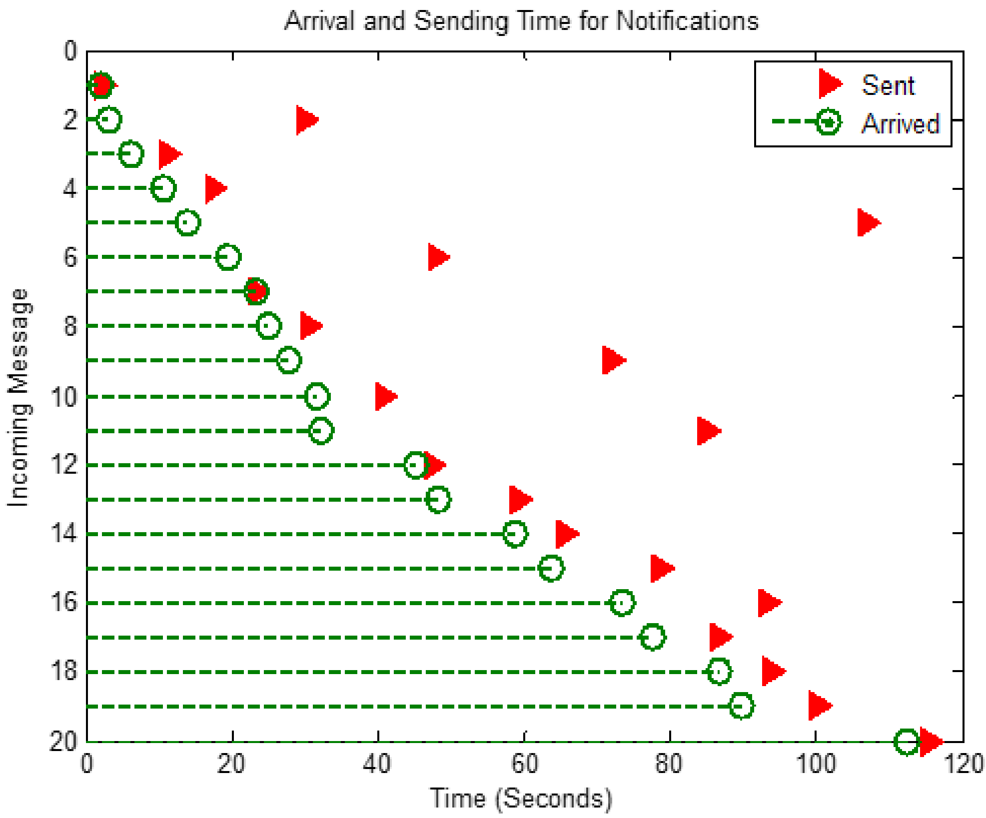

Figure 5.8: Arrival and Sending Time of Notifications

5.2.2 Interval 2

A new message arrives with a mean of 5 seconds

Table 5.6: Incoming Messages and Their Allocations

	# messages	# notifications	# recycling bin	# queue
Priority 1	7	6	1	0
Priority 2	10	6	4	0
Priority 3	7	0	7	0

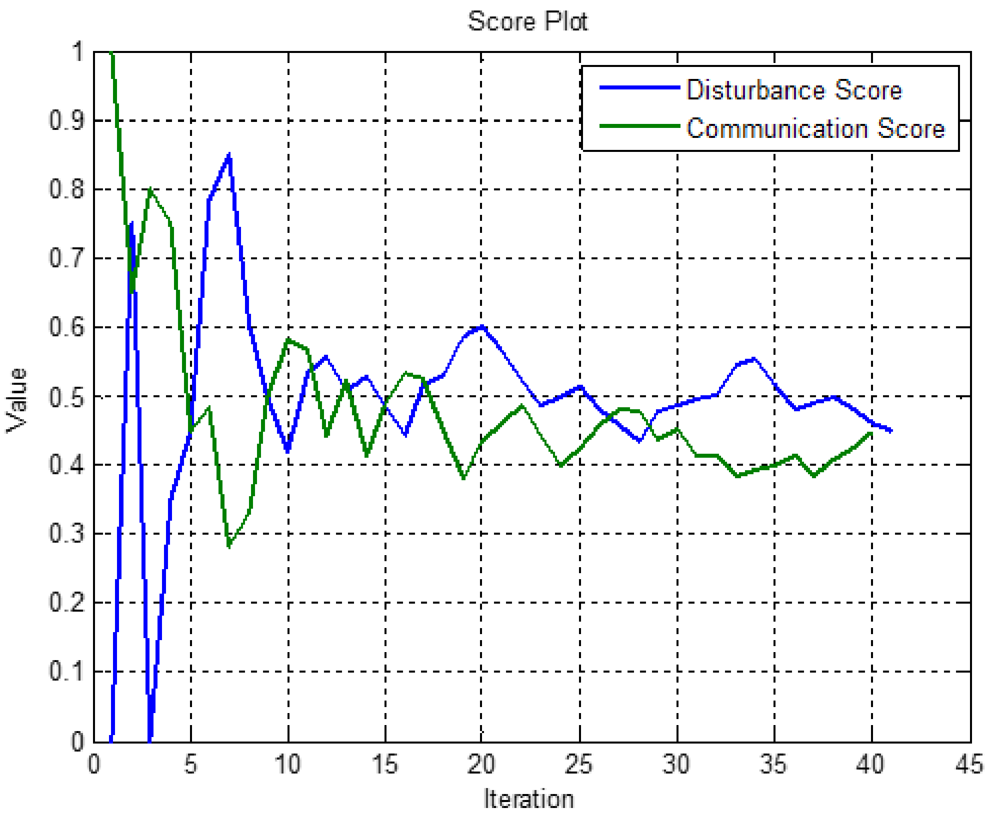

Figure 5.9: Communication and Disturbance Score Development

- Mean delay for a notification = 14.84 sec.
- Standard deviation for delay = 20.45 sec.
- Mean disturbance for a notification= 2.78
- Used modalities: Visual + Haptic (100%)

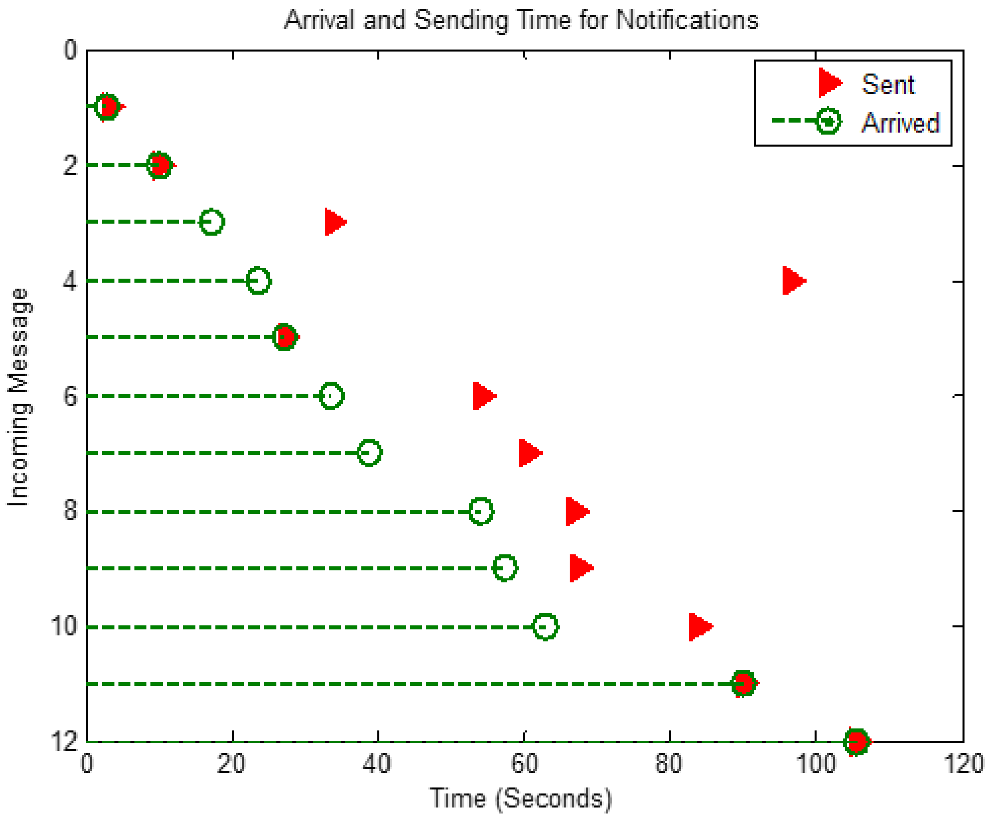

Figure 5.10: Arrival and Sending Time of Notifications

5.3 Scenario 3

Mehmet is walking in the city center, environment is a bit loud and it is daytime. He wants to interact with his smartphone quite a lot so he does not mind if a lot of notifications arrive to his smartphone. User preference ratio $= 1 \longrightarrow \alpha = 1, \beta = 2.6$

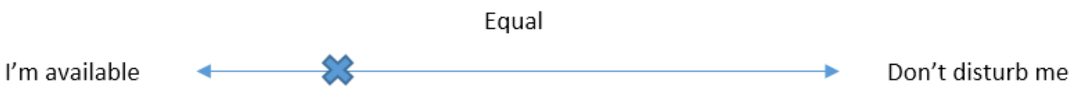

Figure 5.11: Representation of User Preference on 1-D Axis

Table 5.7: User States for Scenario 2

Time	Visual Focus	Illumination	Noise	Speaking	Physical
0 – 60 sec.	Elsewhere	High	Loud	Yes	Walking
60 – 120 sec.	Phone	High	Loud	No	Walking

5.3.1 Interval 1

A new message arrives with a mean of 2.5 seconds

Table 5.8: Incoming Messages and Their Allocations

	# messages	# notifications	# recycling bin	# queue
Priority 1	7	7	0	0
Priority 2	22	17	0	5
Priority 3	15	0	0	15

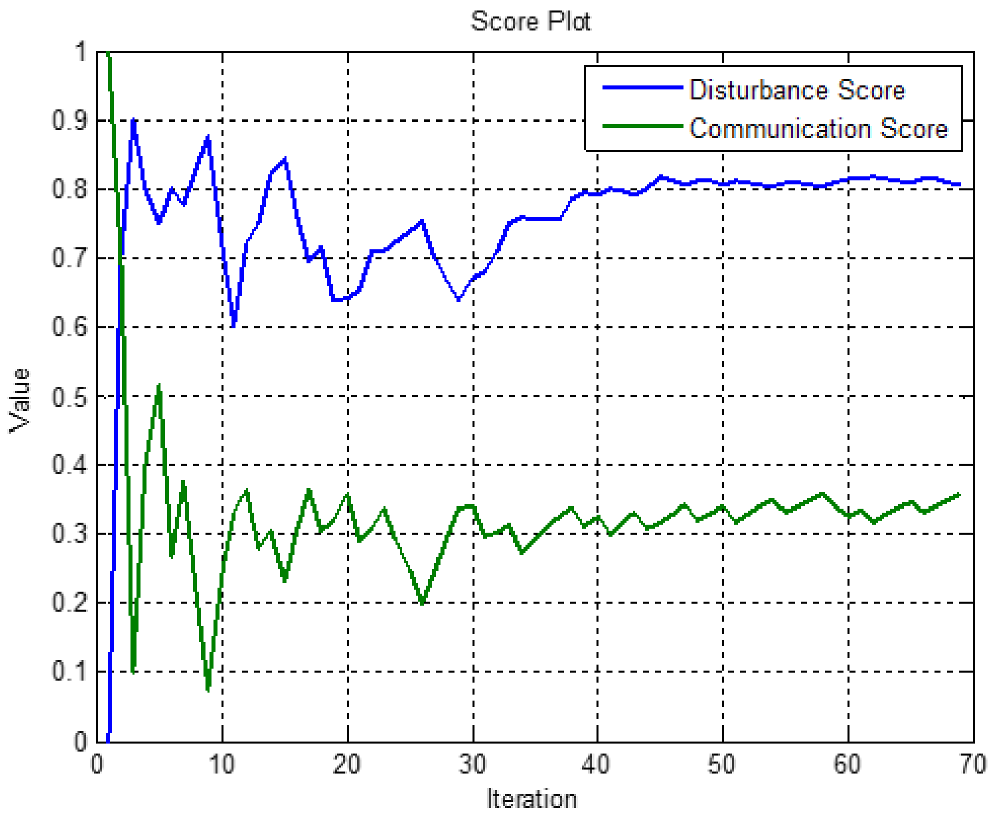

Figure 5.12: Communication and Disturbance Score Development

- Mean delay for a notification = 10.63 sec.
- Standard deviation for delay = 15.21 sec.
- Mean disturbance for a notification= 3.76
- Used modalities: Auditory + Visual (29%) and Auditory + Haptic (71%)

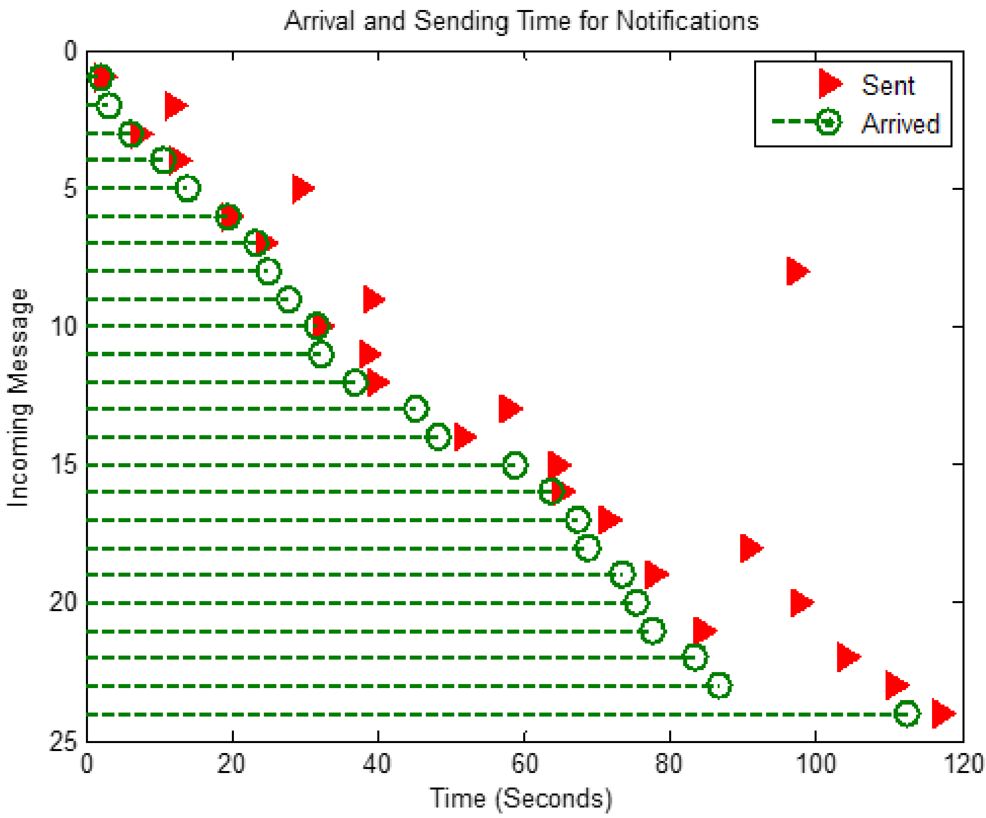

Figure 5.13: Arrival and Sending Time of Notifications

5.3.2 Interval 2

A new message arrives with a mean of 5 seconds

Table 5.9: Incoming Messages and Their Allocations

	# messages	# notifications	# recycling bin	# queue
Priority 1	7	7	0	0
Priority 2	10	10	0	0
Priority 3	7	0	6	1

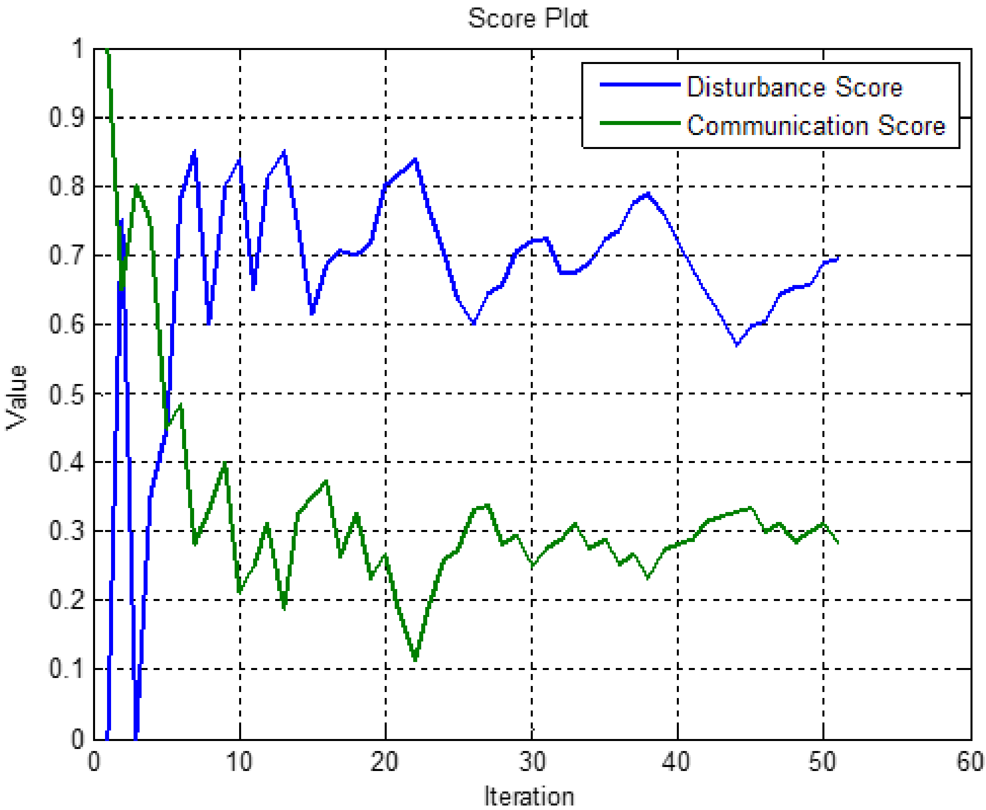

Figure 5.14: Communication and Disturbance Score Development

- Mean delay for a notification = 7.11 sec.
- Standard deviation for delay = 8.58 sec.
- Mean disturbance for a notification= 3.79
- Used modalities: Auditory + Visual (23%) and Auditory + Haptic (77%)

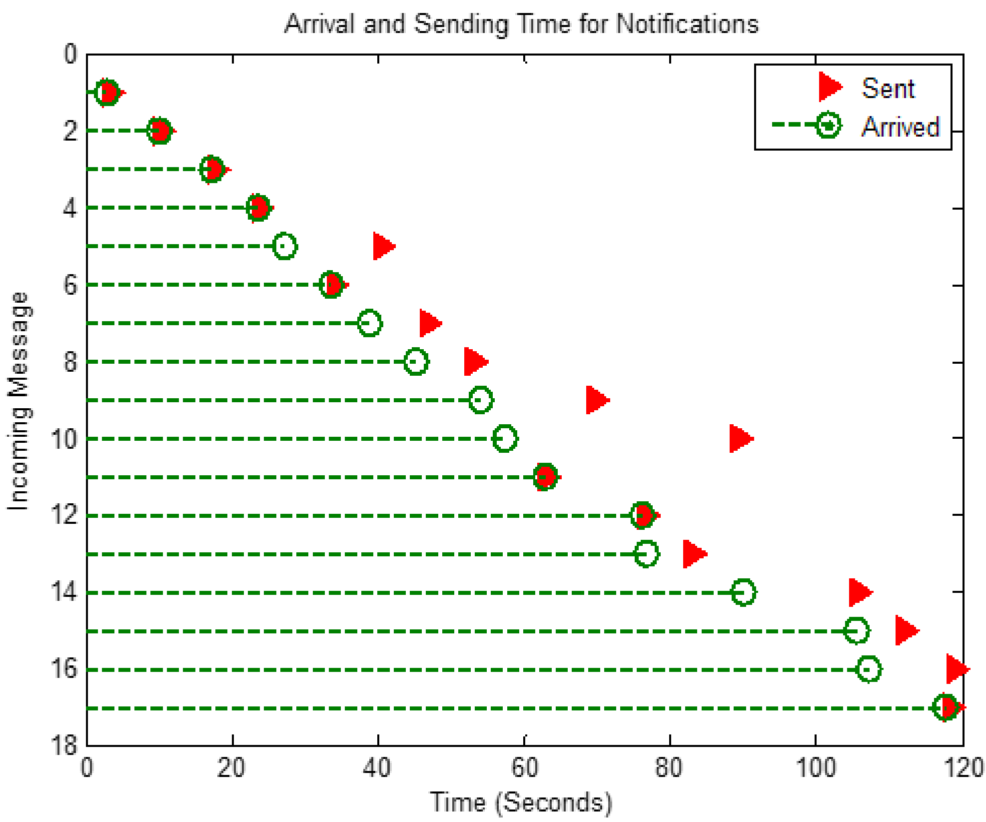

Figure 5.15: Arrival and Sending Time of Notifications

6 Discussion and Conclusion

This work is done to explore and optimize the tradeoff between communication and disturbance in smartphone notification systems. While conducting this research, realistic scenarios, mathematical formulas and statistical information is used in order to define the problem clearly. After the problem definition, a computational approach is used to create a simulation environment to be able to model, test and optimize the delivery mechanism. While preparing the prototype, different scientific methods are implemented such as multiple resource theory, on-line scheduling mechanisms and probabilistic structures. Numerous scenarios are tested in the simulation environment and the results are expressed with graphs and statistics. In this part, the work will be evaluated to assess the compatibility of the results and the research questions.

6.1 Findings

– ***Communication and Disturbance Tradeoff***
  Previous studies [1], [11], [3], [6] claim that sending a notification could either be informative or disruptive and at this point, interruption decision carries considerable weight. Simulations and score graphs show that it is possible to find a stable balance between provisioning worthy information and causing disruption to the user, since user preferences and score levels are always in harmony. However, the system needs a couple of seconds to come to a state of stability since a certain amount of information about the previous decisions are needed, which means some of the first decisions could not be as accurate as later decisions. But since we are talking about a notification system running all the time on a smartphone, this problem appears to be a minor one.

– ***Sending Important Notifications***
  Experiments show that not all the notifications are important [11] and also they might cause attention problems [4]. The number of sent notifications play an important role in keeping these trend health and the limited number of notifications should be decided wisely. As seen from the simulation statistics, notifications with higher priority have a higher rate of being sent to the user, which indicates that the system works accordingly to the scientific information.

– ***Modalities***
  The effect of modality on the notification performance is closely related with the modality selection. It has been shown with experiments that the modality decision has no significant effect on the error rate, however it has a remarkable impact on the response time and the rate of activity in the primary task [24]. Looking from this perspective making the modality decision according to the user's preference on disturbance/communication rate seems rational since the effect of modality is mostly limited to task performance and response rate but not the error rate. On the other hand in such cases like delivering and important information while user does not want to be disturbed might affect the degree of information awareness negatively.

6.2 Performance

– ***Delay***
According to the results, main delay for a notification is strongly correlated with user preference ratio and incoming message density. In each case it is possible to see that more intense incoming message rate and user's preference on being less disturbed affects the mean delay negatively since scheduling problem gets harder. Moreover as seen on the graphs, delay for one notification is mostly not extreme, since standard deviation is always greater than the mean value. This hypothesis is also supported by simulation results and the graph given below, since there are numerous outliners on the normal distribution plot, which means a higher delay value is mostly caused by some outliers. Since there is no strong correlation between immediate attendance and the priority of the notification, and also considering that not all of the notifications prominent[11], delays in the system seem convenient.

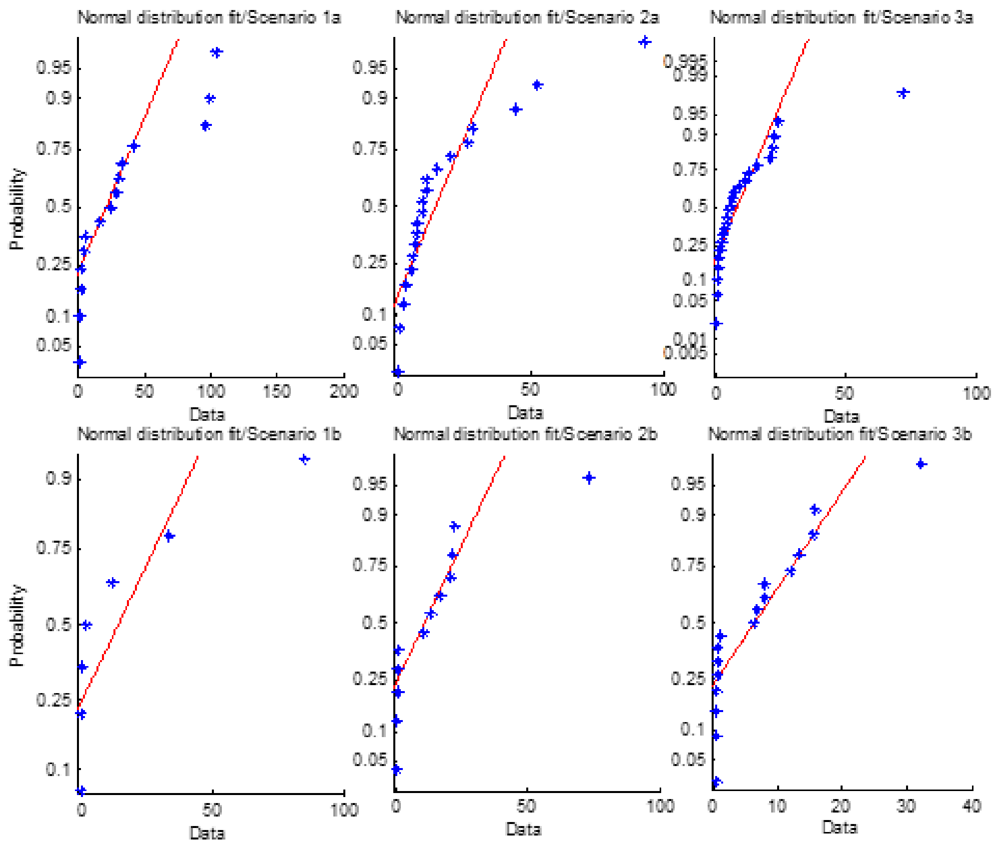

Figure 6.1: Delays for Each Notification and Their Fit to Normal Distribution

– ***Priorities and Decisions***
Simulations and their results show that messages with a higher priority have a greater ratio of being sent to the user with no exceptions, which is exactly what we want. Also user's preference on disturbance/communication rate and the incoming message intensity has a significant effect on the queue occupation, since pushing a notification to the queueing mechanism is highly dependent on those factors. Next figure demonstrates this situation for the previous simulations given in chapter 5.

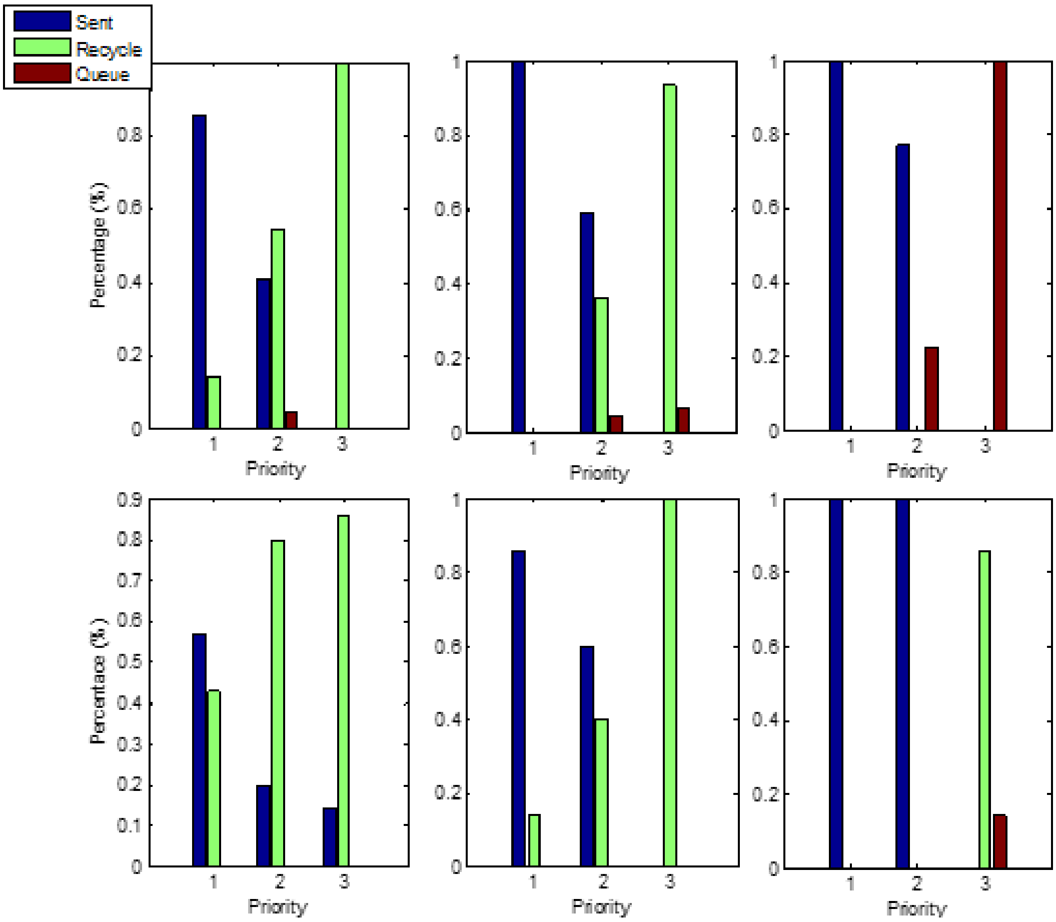

Figure 6.2: Allocation Percentages of Incoming Messages

– ***User preference and disturbance harmony***
As seen from the results, if the user wants to be disturbed less, the computational value of the disturbance is always smaller. Since the user will be more receptive to incoming notifications while she/he is busy with a complex task [1], a smaller disturbance will be enough to get her/his attention and the simulation results are in harmony with this idea.

6.3 Limitations

- ***Attendance to Notifications***
  In this limited simulation environment, the attendance to a notification is not included in the scenario. As pointed out in a previous study [1], it is not very easy to detect the time that user starts to attend the notification thus they assume that user is attended to the notification if the user is already engaged with the smartphone. A study [12] also shows that 70.6% of the notifications are first viewed through the notification drawer, so same strategy is followed in this work, it is assumed that user views the notification with a probabilistic approach depending on the engagement with phone. Looking from this perspective, creating an algorithm with a precise detection of the attendance seems quite complex and this study is limited by this factor.

- ***Modality Decision***
  It is known that the modality selection has an impact on the response ratio to notifications [24] and multiple resource theory [8] suggests that the level of interference has an important effect on task performance. Since the interference calculation relies on a good task analysis to determine the coefficients, task specific cases might not have the same effect on every occasion. Moreover the model's accuracy on resource allocation and interference calculation is limited since human cognitive system is more complex than the model puts forth.

6.4 Future Work

- ***Grouping of Low Priority Messages***
  During the simulations, it is observed that many low priority messages accumulate in the queue and these messages are mostly being sent to recycling bin. A grouping of low priority messages could be done and these messages could be sent as one single notification, e.g. "You have 7 low priority messages", in order to improve the information awareness and the user could decide to see these messages or not.

- ***Participating in Notifications***
  Detecting and considering user's participation to the notifications could provide a better usability. This work considers the viewing of the notification with a probabilistic approach, hence knowing whether user is participating to the notification or not could help in terms of avoiding mistakes to some extend and making on-line scheduling more effective.

– ***How to Deploy?***
To represent the deployment of this idea is demonstrated with sketches. After couple of hours of sketching sessions, it is possible to express the whole idea in one screen. With a simple interface, it is possible to reach all the properties with a single finger stroke. Figure 6.3 is a sample main screen idea for a possible user interface for a smartphone.

- From section 1, switching between different profiles (meeting, studying, sleeping, etc.) and adding new profiles are possible.
- Section 2 provides all the necessary functions to customize a profile by selecting the current profile's availability on a 1-D axis.
- Section 3 includes all the existing application in the smartphone and changing the priority of the app is possible with a single touch.
- Two buttons in the 4th section offers saving and activating the selected profile.

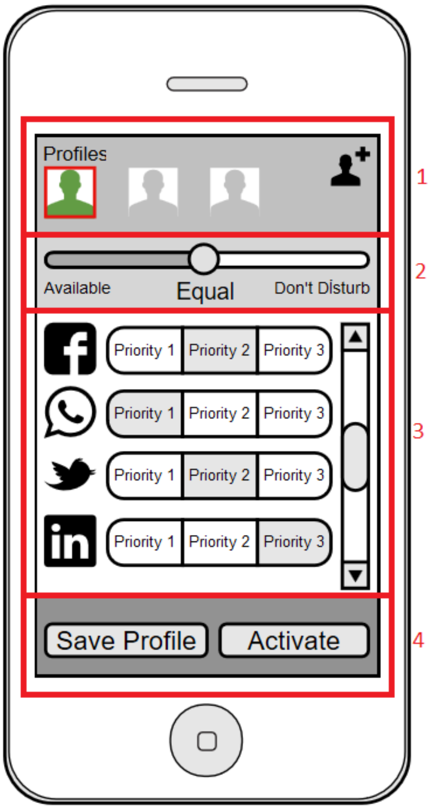

Figure 6.3: A User Interface Idea for Designed Algorithm

6.5 Conclusion

Although many users are suffering from notification systems, this work shows that it is possible to design and implement much better notification algorithms by using decent scientific background. Expressing the problem with scenarios and mathematical formulas, and solving these problems by using scheduling, multiple resource theory and probabilistic approaches made it possible to develop a notification brokering algorithm.

As a product of this long journey, proposed algorithm is able to re-schedule notifications, send messages with high importance as notification with a great ratio difference, use different modalities to cause less task interference and balance the tradeoff between communication and disturbance. Furthermore the work is still open to improvements and this new look to the notification systems could bring great benefits to users in the future.

References

[1] A. Mehrotra, V. Pejovic, J. Vermeulen, R. Hendley, and M. Musolesi, “My phone and me: Understanding people’s receptivity to mobile notifications,” in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’16. New York, NY, USA: ACM, 2016, pp. 1021–1032. [Online]. Available: http://doi.acm.org/10.1145/2858036.2858566

[2] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, “Situation awareness, mental workload, and trust in automation: Viable, empirically supported cognitive engineering constructs,” *Journal of Cognitive Engineering and Decision Making*, vol. 2, no. 2, pp. 140–160, 2008.

[3] D. S. McCrickard, M. Czerwinski, and L. Bartram, “Introduction: design and evaluation of notification user interfaces,” *International Journal of Human-Computer Studies*, vol. 58, no. 5, pp. 509–514, 2003.

[4] K. Kushlev, J. Proulx, and E. W. Dunn, “"silence your phones": Smartphone notifications increase inattention and hyperactivity symptoms,” in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’16. New York, NY, USA: ACM, 2016, pp. 1011–1020. [Online]. Available: http://doi.acm.org/10.1145/2858036.2858359

[5] N. Lavie, “Attention, distraction, and cognitive control under load,” *Current Directions in Psychological Science*, vol. 19, no. 3, pp. 143–148, 2010.

[6] L. D. Turner, S. M. Allen, and R. M. Whitaker, “Interruptibility prediction for ubiquitous systems: Conventions and new directions from a growing field,” in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp ’15. New York, NY, USA: ACM, 2015, pp. 801–812. [Online]. Available: http://doi.acm.org/10.1145/2750858.2807514

[7] S. LeeTiernan, E. Cutrell, M. Czerwinski, and H. Hoffman, “Effective notification systems depend on user trust,” in *Proceedings of Human-Computer Interaction–Interact*, 2001, pp. 54–69.

[8] C. D. Wickens, “Multiple resources and performance prediction,” *Theoretical Issues in Ergonomics Science*, vol. 3, no. 2, pp. 159–177, 2002. [Online]. Available: http://dx.doi.org/10.1080/14639220210123806

[9] A. Zalzala and P. F. Chapter, “7: Job-shop scheduling (pp. 134–160),” *Genetic algorithms in engineering systems. Edited by. IEE The Institution of Electrical Engineers control engineering series*, vol. 55.

[10] M. Harchol-Balter, N. Bansal, B. Schroeder, and M. Agrawal, “Srpt scheduling for web servers,” in *Job Scheduling Strategies for Parallel Processing*. Springer, 2001, pp. 11–20.

[11] A. Sahami Shirazi, N. Henze, T. Dingler, M. Pielot, D. Weber, and A. Schmidt, "Large-scale assessment of mobile notifications," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '14. New York, NY, USA: ACM, 2014, pp. 3055–3064. [Online]. Available: http://doi.acm.org/10.1145/2556288.2557189

[12] M. Pielot, K. Church, and R. de Oliveira, "An in-situ study of mobile phone notifications," in *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services*, ser. MobileHCI '14. New York, NY, USA: ACM, 2014, pp. 233–242. [Online]. Available: http://doi.acm.org/10.1145/2628363.2628364

[13] R. Hansson, P. Ljungstrand, and J. Redström, *Ubicomp 2001: Ubiquitous Computing: International Conference Atlanta Georgia, USA, September 30–October 2, 2001 Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, ch. Subtle and Public Notification Cues for Mobile Devices, pp. 240–246. [Online]. Available: http://dx.doi.org/10.1007/3-540-45427-6_20

[14] J. Pasquero, S. J. Stobbe, and N. Stonehouse, "A haptic wristwatch for eyes-free interactions," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 3257–3266. [Online]. Available: http://doi.acm.org/10.1145/1978942.1979425

[15] S. Garzonis, S. Jones, T. Jay, and E. O'Neill, "Auditory icon and earcon mobile service notifications: Intuitiveness, learnability, memorability and preference," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '09. New York, NY, USA: ACM, 2009, pp. 1513–1522. [Online]. Available: http://doi.acm.org/10.1145/1518701.1518932

[16] S. T. Iqbal, X. S. Zheng, and B. P. Bailey, "Task-evoked pupillary response to mental workload in human-computer interaction," in *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '04. New York, NY, USA: ACM, 2004, pp. 1477–1480. [Online]. Available: http://doi.acm.org/10.1145/985921.986094

[17] V. S. Pai, P. Druschel, and W. Zwaenepoel, "Flash: An efficient and portable web server." in *USENIX Annual Technical Conference, General Track*, 1999, pp. 199–212.

[18] M. Crovella, R. Frangioso, and M. Harchol-Balter, "Connection scheduling in web servers." in *USENIX Symposium on Internet Technologies and Systems*, vol. 10, 1999, pp. 243–254.

[19] W. Mao, R. K. Kincaid, and A. Rifkin, *The Impact of Emerging Technologies on Computer Science and Operations Research*. Boston, MA: Springer US, 1995, ch. On-Line Algorithms for a Single Machine Scheduling Problem, pp. 157–173. [Online]. Available: http://dx.doi.org/10.1007/978-1-4615-2223-2_8

[20] D. Chen and R. Vertegaal, "Using mental load for managing interruptions in physiologically attentive user interfaces," in *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '04. New York, NY, USA: ACM, 2004, pp. 1513–1516. [Online]. Available: http://doi.acm.org/10.1145/985921.986103

[21] P. Tarasewich, C. S. Campbell, T. Xia, and M. Dideles, *UbiComp 2003: Ubiquitous Computing: 5th International Conference, Seattle, WA, USA, October 12-15, 2003. Proceedings.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, ch. Evaluation of Visual Notification Cues for Ubiquitous Computing, pp. 349–366. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-39653-6_27

[22] P. A. Hancock, T. Oron-Gilad, and J. L. Szalma, "Elaborations of the multiple-resource theory of attention," *Kramer, AF, Wiegmann, DA„ K.(eds.) Attention: From Theory to Practice*, pp. 45–56, 2006.

[23] W. J. Horrey and C. D. Wickens, "Multiple resource modeling of task interference in vehicle control, hazard awareness and in-vehicle task performance," in *Proceedings of the Second International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design*, 2003, pp. 7–12.

[24] D. Warnock, M. McGee-Lennon, and S. Brewster, *Human-Computer Interaction – INTERACT 2011: 13th IFIP TC 13 International Conference, Lisbon, Portugal, September 5-9, 2011, Proceedings, Part II.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, ch. The Role of Modality in Notification Performance, pp. 572–588. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-23771-3_43

[25] S. K. Card, T. P. Moran, and A. Newell, "The keystroke-level model for user performance time with interactive systems," *Commun. ACM*, vol. 23, no. 7, pp. 396–410, Jul. 1980. [Online]. Available: http://doi.acm.org/10.1145/358886.358895