

Attention and Intention Goals Can Mediate Disruption in Human-Computer Interaction

Ernesto Arroyo¹ and Ted Selker²

¹ Department of ICT, Universitat Pompeu Fabra,
Barcelona, Spain

ernesto.arroyo@upf.edu

² Carnegie Mellon Silicon Valley
Moffett California, USA

ted.selker@sv.cmu.edu

Abstract. Multitasking environments cause people to be interrupted constantly, often interfering with their ongoing tasks, activities and goals. This paper focuses on the disruption caused by interruptions and presents a disruption mediating approach for balancing the negative effects of interruptions with respect to the benefits of interruptions relevant to the user goals. Our work shows how Disruption Manager utilizing context and relationships to user goals and tasks can assess when and how to present interruptions in order to reduce their disruptiveness.

The Disruption Management Framework was created to take into consideration motivations that influence people's interruption decision process. The framework predicts the effects from interruptions using a three-layer software architecture: a knowledge layer including information about topics related to the ongoing activity, an intermediate layer including summarized information about the user tasks and their stages, and a low level layer including implicit low granularity information, such as mouse movement, context switching and windowing activity to support fail-safe disruption management when no other contextual information is available. The manager supports implicit monitoring of ongoing behaviors and categorizing possible disruptive outcome given the user and system state. The manager monitors actions and uses common sense reasoning in its model to compare communication stream topics with topics files that are active on the desktop.

Experiments demonstrate that disruption manager significantly reduces the impact of interruptions and improve people's performance in a multi-application desktop scenario with email and instant messaging. In a complex order taking activity, disruption manager yielded a 26% performance increase for tasks prioritized as being important and a 32.5% increase for urgent tasks. The evaluation shows that the modulated interruptions did not distract or troubled users. Further, subjects using the Disruption Manager were 5 times more likely to respond effectively to instant messages.

Keywords: Disruption, Interruption, Adaptive Interface, Software Managers, Human Computer Interaction.

1 Introduction

The increase of information overflow and continuous request for user attention [20] makes interruptions a common occurrence in computing environments. Users might delegate another person or a computerized intelligent agent to perform tasks on their behalf to avoid cognitive overload and successfully perform multiple tasks. With enough delegation, management of it can also cause interruption [27]. This paper focuses on the disruption caused by interruptions presented to users by computer technologies as the result of a request for synchronous or asynchronous communication. Careful design of interrupting notifications might help reduce disruption effects on people's performance on ongoing activities. But in order to keep up with the ever-increasing demands for user attention, future computers environments will require active mediators that are able to interpret and recognize the value of communication. Indeed, mediating approaches have been helpful in developing successful software agents such as spam filters that act on behalf of the user. Our work deepens such systems into the area of mediating people's communications with friends and colleagues so that people are able to maintain concentration amidst their busy different lifestyles. While it is possible to restrict all interruptions so that they do not disrupt the ongoing task, sometimes interruptions represent important corollary tasks, opportunities you never want to pass up, or being responsible for tasks of even a higher priority such as health. We argue that user goals and motivations should take precedence over micro-task benefits in considering which interruptions to block or attempt to block. To illustrate this argument, we present a Disruption Management Framework designed as a modeling method for designing computer-based disruption managers. The framework outlines some of the factors needed to mediate disruption in computing activities regarding the interruption context and its relationships to the user's goals.

1.1 Interruptions, Disruption, and Distraction

Interruptions are an everyday and normal part of human behavior. People frequently interrupt communication dialogue, such as an unanticipated request for topic switching while having a conversation [29]. An *interruption* can be defined as an unanticipated request for task switching from a person, an object, or an event while multitasking [29][30][24][15]. Interruptions typically *request immediate attention* and *insist on action* [10], and reduce productive focus. If an interruption is allowed to distract the user into action, it escalates into *disruption* [23]. Thus, disruption is defined as the negative effects on a primary task from interruptions requiring transition and reallocation of attention focus from one activity to another. This paper will use the term disruption to accommodate a situation in which a request has been accepted from another task, causing a negative effect on the ongoing activity. Although the main focus of this paper is disruption it is important to mention that distractions and interruptions are similar in that they can both occur while the user performs a primary activity. For instance *distraction conflict theory* represents a research stream investigating how distractions can be ignored or processed at the same time as the primary activity [17]. Unlike distractions, interruption-disruption shares the same sensory channel as the primary activity and encompasses a task that

could be performed. Our main interest is in on interruptions that result in capacity and structural interference and that disrupt the ongoing activity; negatively affecting human performance [18].

1.2 Interruption Management

Coordinating interruptions involves one or more person's modes of activity: cognition, perception, or physical action. Interruption coordination is defined as the method by which a person shifts their focus of consciousness from one processing stream to another [29]. How efficiently and effectively interruptions are handled by users might depend on characteristics of the notification itself and characteristics of the ongoing activity [7]. However, people's reactions to interruptions and perceived disruption are principally affected by goal oriented strategies adopted to evaluate incoming interruptions in order to accomplish their goals [24], [28], [5]. For example, a person who works at an office is more likely to take an incoming phone call from a co-worker while at work than when he is on his way home. It is common for people to juggle several competing goals at one time, while their priorities might change depending on various factors. This is exemplified in a diary study of office work, which reported frequent and deliberate task-switching activities [14]. Similarly, residential interviews and self-reports revealed that willingness to handle interruptions varies across individuals with current location, as well as with current activity [31].

Can a computing system recognize and mediate relative to the underlying factors that influence people when dealing with interruptions? Our work addresses this question and shows how a disruption manager utilizing context and relationships to user goals and tasks can reduce disruptive qualities of interruption requests.

1.3 Disruption Management

One of the key questions for the understanding of human disruption is identifying the factors that play a role in people's decision process regarding interruptions. Previous research has focused on identifying task complexity and its influence on user performance [4][11], the coordination method used to handle interruptions [29], the interruption point at which interruptions arrive [11][21], the similarity between ongoing and interrupting tasks [15], and the interruption modality [24]. We expand this to focus on the factors that influence people's decision process at turning interruptions into actual disruption. A factor strongly correlated to how people react to interruptions is the level of goal commitment, that is, the importance of the task and the belief that the goal can be accomplished [26] [28]. The importance of the goal to the individual, will affect subsequent reaction to an interruption.

We explore the hypothesis that people's reactions to interruptions and perceived disruption are principally affected by goal-oriented strategies users adopt to evaluate incoming interruptions in order to accomplish their goals. We argue that incoming interruptions are evaluated with respect to the ongoing processes goals and priorities. Therefore, it is possible for people to be influenced by the level of commitment to a task [21][7][36]. If a task is almost completed, people can opt for finishing the task before accepting an interruption and switch to the interrupting task as task priorities are constantly changing.

2 Disruption Mediation Framework

Disruption Mediation Framework (DMF) introduces a multilayer software model that separates important constituents of disruption and aids in creating a computer-based disruption manager. Earlier interruption models [24][32][1][20][25] have not supported continuously monitoring the user and dynamically adapting timing and interrupting modality, among other parameters. Indeed, current interruption models focus on the user, disregarding the context from interrupting applications. Based on our work and existing research we take the approach that goal concepts and task context serve as important factors in predicting disruption. The DMF is influenced by existing literature, and is especially an extension of our explorations in three areas: tracking user activity, identifying ways of tracking ways to interrupt people, and understanding the factors that influence the interruption decision process (task completion level, task type, task complexity, task switching strategies, etc.) [3][4][12][22][35].

The DMF identifies a decision process that evaluates incoming interruptions with respect to the current state of the ongoing activity or situation. It determines the priorities of each interruption to decide when to execute the task associated with the interruption. Previous work investigating interrupting modalities has shown that the decision process is affected by people's individual differences, such as prior experiences, motivations, and psychological factors [1][15][19]. The DMF also identifies goal priorities, goal commitment, and goal relevancy as other factors affecting the interruption-decision process.

The DMF introduces **goals** and relevance (to these goals) as a main factor for mediating interruptions. The interruption process is closely related to attention (according to the information-processing model). Attention also determines which goal concepts are relevant. These concepts then can also determine which tasks are important or have higher priority. For this reason the DMF uses concepts to provide a cognitive representation and offer insight into the user's attention. The DMF also includes, **user activity**, and **task context** as other important factors that allow examining incoming interruptions in order to evaluate their potential for disruption (see **Fig. 1**). Our approach is meant to be usable and easily implemented for any interactive digital system.

2.1 Relationships between Goals and Concepts

SuwatanaPongched [34] classified interruptions into three categories: 1) Interruptions relevant to the primary task that assist completion of the primary task. 2) Interruptions irrelevant but related to the primary task, although not contributing to completion of the primary task. 3) Interruptions irrelevant and unrelated to the primary task. These interruption categories are useful; however, people often perform more than one task at a time in order to accomplish their goals as several tasks can be grouped as being part of a single goal. The above interruption classification schema can be extended to include the relationship to the user's goals. That is, interruptions irrelevant to the primary task can be related to the user's goal, thus contributing to completion of one of the user's goals. Some of these goals are unique in that the user might be willing to sacrifice a certain amount of primary task attention in order to achieve them. Literature agrees that irrelevant, unrelated interruptions can be harmful to the primary task, and that they elicit frustration and anxiety [11][13][15][29]. Our work has also shown that people can benefit from interruptions if they are relevant to the ongoing task or the user's goals [35].

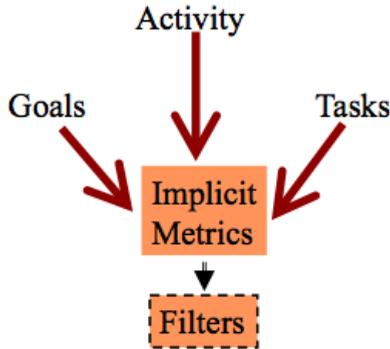


Fig. 1. The Disruption Management framework (DMF) identifies goal concepts, user activity, and task context: the three main components to mediate and filter disruption from implicit metrics

Current approaches for reasoning about the user’s goals have focused on sensing user actions directed at achieving a domain-specific goal [16]. This approach is dependent on previous examination of the desired domain and is limited to known domains or the domain itself. Instead, the DMF uses concepts such as topic of documents, and communication derived from the user environment as a way to reason and think about the underlying user’s goals. Tools such as, WordNet¹ and ConceptNet² can support context-oriented inferences over real-world texts [33], by supporting semantic similarity computations and by performing query expansion from a given concept. These semantic knowledgebase systems make reasoning about goals through concepts a practical approach.

2.1 Activity Monitoring

The DMF focuses on supporting people’s goals as a mean to reduce disruption. The degree to which interruption mediated interfaces support the user goals is a key factor that determines their success. The success is also determined by inferences generated from interpreted knowledge; we call virtual sensors [25]. Thus, The goal is to identify “sensors” that generate sufficient knowledge independent from domain-specific sources. These domain-independent sources provide the basis for mediating disruption when no other data sources are available. Low-level factors, such as micro tasks and HCI interactions (mouse and keyboard behavior, windowing activity, etc.) can be used for mediating interruptions, as they provide a fail-soft system response when no domain-specific data is available.

2.3 Tasks

The benefit from accepting interruptions should be balanced with respect to the ongoing task. The challenge consists of balancing interruptions at a task level while

¹ Princeton University "About WordNet." <http://wordnet.princeton.edu>

² ConceptNet <http://csc.media.mit.edu/conceptnet>

supporting the user’s goals. Our approach extends and makes use of some constructs defined by previous research in the area of interruptions. This research indicates that disruption is related to several task factors: 1) Task complexity and similarity of the primary or interrupting task [15][22], 2) Interruption relevancy to the primary task, task stage when interrupted [13], 3) Interruption coordination method [29], 4) Modalities of the primary task and interruption [2][24]. In our framework task context includes *Number of Tasks, Task level, Task Type, Task Complexity, Task Difficulty, and Task priority.*

3 Disruption Manager

We developed a test-bed to evaluate dynamic interruption systems. The test-bed allows the examination of the relation between ongoing behaviors, task actions, goals and interruptions. The application is implemented as a three-layer architecture system (see Fig. 2). A low-level layer includes implicit low granularity information such as keystrokes and mouse movement activity. An intermediate layer includes the activities and information to which some of the low granularity data can be extracted and summarized, such as reading, switching tasks, paying attention, etc. And a top layer or knowledge layer includes the information or concepts related to the user goals.

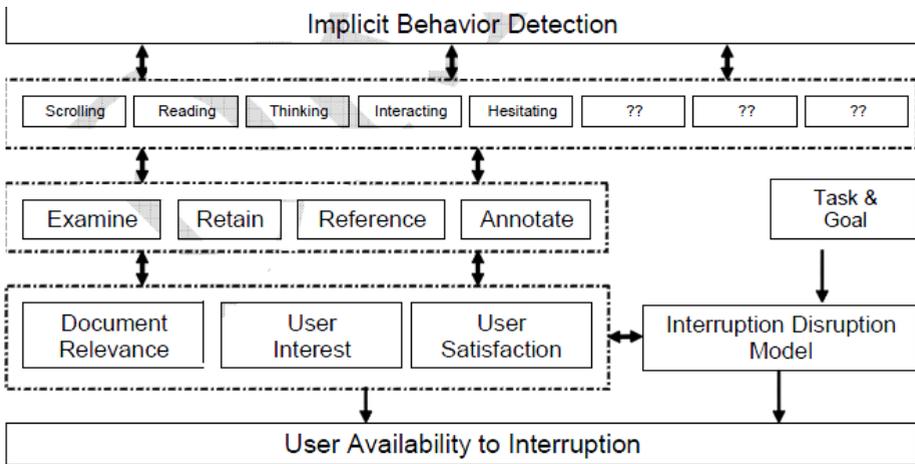


Fig. 2. Disruption manager three-layer architecture: A low level layer includes implicit low granularity information (top). An intermediate layer includes the user activities and information to which some of the low granularity data can be extracted and summarized (middle-left). And knowledge layer includes the information or concepts related to the user goals (right).

The disruption manager monitors the user state (current activity), concepts surrounding the user’s goals: history of recently accessed documents, web pages and search queries, the interrupting message relevance to these concepts, and concept priority. The manager then identifies messages that should be allowed to interrupt the user or delayed to an appropriate time within task execution. The disruption manager

uses monitoring modules to track the user state, concepts surrounding the user's goals and interrupting message concepts. The system has one module for each context (goals, tasks, and activities) in which the interruption content is examined and a decision module that mediates interruptions on multiple services based on the evaluations of the context modules. The mediator uses several auxiliary modules for interfacing with an Instant Messaging client to both read in the Instant Message (IM) content and manipulate the timing and presentation of the IM. A context module that uses natural language on the goal-level layer; experience, interest, and reading for the intermediate layer; and mousetracking, task-stage, and task-exit for the low-level layer. These architectural elements in disruption manager are described in the following section.

3.1 Context Modules

Each Context Module is responsible for evaluating a particular aspect of the interruption, the system, and the user. These modules are derived from aspects of the disruption model. The modules convey their evaluations to the Decision Module as appropriateness of showing the IM at a given moment. The manager's top level monitoring layer uses Google Desktop³ and ConceptNet [37] engines as services running on the user's computer. Google Desktop keeps an up-to-date index of files and documents and their contents. ConceptNet is a commonsense knowledgebase with facts from the Open Mind Commonsense corpus [33]. Its concise semantic network contains 200,000 assertions and supports practical textual-reasoning tasks over real-world documents.

Natural Language Module. (NL) implements the part of the disruption module concerning the relatedness of the content of the IM to documents the user is working with. It uses natural language processing and commonsense reasoning to develop an understanding of the interruption and documents, and attempts to compare the interruption to each document. These comparisons are aggregated into a relevance score, indicating the relevance of the IM to all documents examined. A separate score focuses on relevance to the current document.

The NL Module has three major components. The first is responsible for obtaining the text of an IM (this is done through the IM Interface).

The second component is responsible for locating files of interest and obtaining the contents of those files. The manager application queries the Google Desktop Engine for recently accessed documents, files of interest (PDF, DOC, PPT, etc), emails, IMs and web pages and parses them using ConceptNet and a natural language processing engine. Files of interest are the files open on the user's computer, as well as recently viewed documents and webpages. The system also obtains a list of open files using VBScript and the Microsoft PsTools⁴ library. The system then uses Google Desktop [39] to locate and read those files and to search for web pages viewed in the past hour in the web cache, and to find and read the documents in the user's My Recent Documents folder.

The third component uses document-level functions in ConceptNet (text normalization, commonsense-informed part-of speech tagging, semantic recognition, chunking, surface parsing, thematic-role extraction, and pronominal resolution) to extract

³ Google Desktop: <http://desktop.google.com/>

⁴ PsTools <http://www.microsoft.com/technet/sysinternals/utilities/pstools.mspx>

the verb-subject-object-object frames from recently accessed documents. The entire contents of both the IM and all of the retrieved documents are individually fed into the MontyLingua⁵ natural language processing suite. The MontyLingua suite provides both lexical parsing of text and commonsense reasoning through the OpenMind⁶ commonsense database. The NL Module extracts from MontyLingua interpretation key words and concepts in the texts, uses a thesaurus to find possible synonyms for those words and concepts, and then counts the number of times the important words, concepts, and synonyms from the IM appear in the other documents. The NL module then extracts all the concepts in a document, assigns them saliency weights based on lightweight syntactic cues, and computes their weighted contextual intersection.

Concept connections in ConceptNet's semantic network allow the contextual neighborhood around a concept to be found by spreading activation radiating outward from a source concept node. The more frequently the number of important words or concepts appears, the more relevant the content of the IM is likely to be. The output of the NL module is the average number of times a key word or synonym in the IM appears per sentence in all the searched documents.

This module allows the manager to summarize text of active documents, identify the documents gist topics, evaluate notifications, capture and classify incoming messages, detect if actions are required from the user, keep track of topics related to ongoing and past goals, and determine if incoming interruptions should be presented to the user.

Mouse-tracking Module (MT). This MT module is composed of three components and represents the portion of the disruption module concerning the user's familiarity with and their depth of involvement in their task. This component records the user's mouse movements and reasons about the user by comparing his or her mouse movements to a database of many people's mouse movements. The Mousetracking Module serves to determine the experience of a user with a website, the interest level of a user in a website, and whether the user is reading or scanning a website. This module is based on experimental data monitoring mouse activity while browsing websites [8].

The manager's low-level proxy-based layer installed on the user's computer monitors and categorizes mouse movement activity into low granularity behaviors (scrolling, menu, text input, clicking) and user states (reading, deciding, scanning, and waiting). The manager uses a naïve Bayes classifier and wrapper based feature selection because it is computationally inexpensive, which is important when running on a web-browser due to its limited resources.

The mouse tracking module outputs data representing the percent experience, the percent interest, and the percent likelihood the user was closely reading a webpage. The Decision Module uses these heuristics to estimate the depth of user involvement with their current task, with the idea being that the more deeply involved a user is with their current task, the more costly it is to interrupt the user.

Task Stage Module (TS). The TS module is responsible for determining if a user is at the beginning, in the middle, at the end, or between tasks. It does this by looking for discontinuities or changes in keyboard and mouse usage and windowing behavior. This is used to determine the user cognitive load based on HCI interactions. The TS

⁵ Montylingua: <http://web.media.mit.edu/~hugo/montylingua/>

⁶ OpenMind: <http://www.openmind.org>

looks for significant changes to the number of keystrokes per minute, mouse time per minute, or windowing behavior indicating that the user is changing tasks, or subtasks, as at those moments interruptions are more likely to be less disruptive [21].

Existing Tasks Module (ET). The ET module attempts to gain an understanding of persisting tasks the user may have, even though they are not currently working on them, and corresponds to the part of the disruption module which determines if interruptions relate to other tasks the user has but may not be currently working on. The ET module returns the percentage to which an interruption is appropriate for past tasks. This information is a heuristic for how often the user cares about what the interruption is about, and thus also on how significant tasks related to the interruption are to the user.

Decision Logic Module (DL). The DL module is the central component of the disruption manager. Whenever an IM arrives, DL determines the appropriateness of that message. It polls all Context Modules for their evaluations of the IM, and decides how to proceed. Once the disruption manager decides an interruption should be presented, it delays the interruption until an appropriate time in order not to disrupt the ongoing micro-task or activity. If the interruption is relevant to the user’s goal, the manager gives priority to this interruption, and presents it as soon as possible; while minimizing disruption on the ongoing task. Delaying standards are lowered over time to guarantee the message will be displayed within 2 minutes.

In addition, the manager’s decision rules are based on findings showing that interruptions related to topics the user has worked on in the past, have the potential to be significant to the user’s goals and should be allowed [5][22][28]. Thus, the manager limits the number of unrelated interruptions in order to reduce perceived disruption. The manager also limits interruptions whenever this ratio increases and allows interruptions related to prioritized topics whenever confidence values are above predefined thresholds.

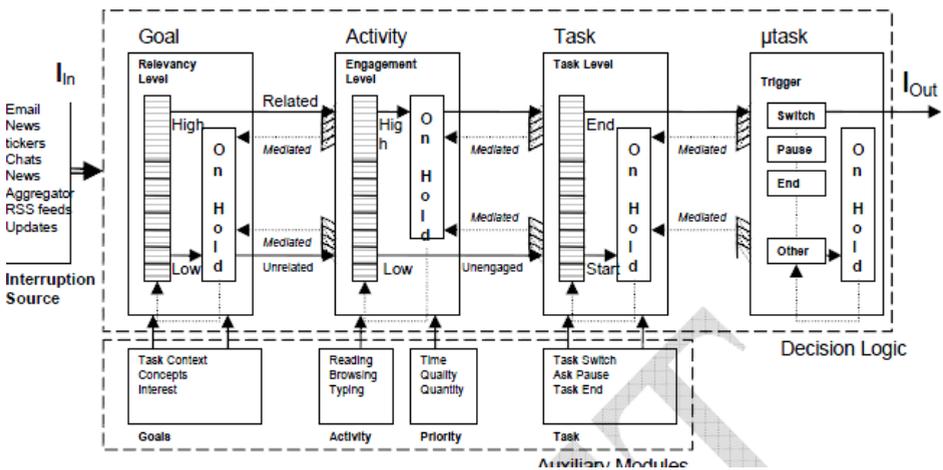


Fig. 3. Disruption manager’s Decision Logic Module: a layered filtering process determining the stages that each interruption must go through before being delivered to the user

The decision module for the disruption manager is implemented as add-on to Trillian™; a stand-alone chat client that supports AIM, ICQ, MSN, Yahoo Messenger, and IRC. This allows the disruption manager to be easily deployed and integrated into current systems without requiring the user to lose existing contacts, learn a new interface. The IM client provides unique customization functionality, such as, contact message history, and an advanced automation system to trigger events based on anything that happens in the client. This allows the disruption manager to “catch” incoming interruptions and control them (see **Fig. 3**).

4 Evaluation

An experiment evaluated Disruption Manager’s effectiveness in mediating interruptions compared to existing interfaces. The experiment elicits disruption by presenting interruptions while the user is engaged on a task. These interruptions present themselves as notifications that carry a message associated with them, requiring the user to switch to another task. In addition, the message associated with each interruption can either be related or unrelated to the user goals.

The experiment evaluated Disruption Manager with respect to *productivity* and user *satisfaction*. Productivity refers to objective performance metrics such as overall goal completion, the time taken to finish an activity, task or goal. While user satisfaction refers to subjective metrics designed to evaluate the perceived user satisfaction for given tasks and overall goals. The main dependent variables were performance and perceived disruption. Several other variables were used to confirm the task was performed properly. These variables included task time, number of notifications attended to, time spent on each email, and STAI⁷ score.

4.1 Hypothesis

1. People using Disruption Manager will have higher performance than people in the No-Manager condition.
2. People using Disruption Manager will be more efficient in their task than people in the No-Manager condition.
3. People using Disruption Manager will report less perceived disruption than people in the No-Manager condition.

4.2 Method

The effectiveness of Disruption Manager was assessed using a between subjects experimental design. Subjects were randomly assigned to one of two conditions: 1) Communications mediated by Disruption Manager or 2) Communications presented normally as the IM application receives them (No-Manager). In the Disruption Manager condition, the manager controls email notifications presented based on whether the email is related to the ongoing activity and several factors. **Fig. 3** shows the filtering stages that each interruption must go through before being delivered to the user. The manager allows people to complete the task without unnecessary

⁷ State-Trait Anxiety Inventory for Adults.

distractions. That is, related IMs are presented (almost) right away so that the subjects can benefit from the IM. On the other hand, irrelevant IMs are delayed until a subtask is finished. The manager's behavior can be summarized with the following rules:

- Relevant IM are presented after small changes in activity, such as quick task switches, or after finished finding an item, updating values, text entry, etc.
- Irrelevant IM are presented after subjects finish gathering data for one customer, or finish sending email.
- Allow Instant Message notifications if relevant to current email /customer request (active email, document, or webpage).
 - Relevant presented almost immediately.
 - Wait until finished task or task switch.
- Delay Instant Message notifications if relevant or moderately relevant to current email /customer request (active email, document, or webpage)
 - Wait until a task break.
 - Wait for a task switch only
- Delay Instant Message notifications if not relevant to current email /customer request (active email, document, or webpage).
 - Wait until email sent or task completion.

User Task Scenario. The scenario consisted of customer service and order processing activity for an e-commerce site. We simulated a typical small business environment where customer service representatives take email orders from several customers and process each order individually trying to satisfy the customer's demands and complete a sale. This represents the most commonly reported task for 77 million workers who used computers at work in 2003 according to a Survey by the USA Bureau of Labor and Statistics [6]. Subjects were told that customer service representatives obtain a commission based on their sales and instructed subjects to play the role of a customer representative. Adding this role guaranteed that subjects would perform the task to the best of their abilities and encouraged subjects to obtain a bigger profit.

The customer service scenario serves as a good proxy for representing high-level goals. Rather than specifying detailed task-based goals, the scenario presents an abstract goal requiring subjects to create their own definition of the goal based on the task constraints. The abstract task goal is to satisfy the customer demands based on their requirements. This type of settings addresses the situation where a system might not be able to identify the exact representation of the user's goals, and relies instead on the specific concepts surrounding the user's goals. Subjects were instructed that the experimental setup evaluated an automated system that filtered customer e-mails and an Instant Message (IM) system for price updates, and requests from fellow employees. Centering the experiment on the email manager system focused subjects' attention on the email content, rather than on their own reactions to IM. This warranted that people would react to interruptions, as they would normally do.

User Task Description. The tasks presented in the scenario are similar to those used in a customer service task done at a large supply company in the purchasing department. Small to medium e-commerce sites still require human intervention and process customer orders in a similar way. The main task is focused on processing email orders; the task

includes different types of customers, and the importance of satisfying customer's requests. The tasks are controlled by two systems for improving productivity: 1) An automated system that assigns different type of customers to different employees through the day, as to maintain a balanced workload. The system classifies and sorts customer emails depending on the type of service requested and the customer's demands (high accuracy, fast response, high volume, and low-volume clients). 2) An instant messaging system that allows its employees to share pricing information with one another. Sharing information benefits the company and participants receive a bonus based on the company performance.

User Task details. Each order included a short email script where customers described which items they wanted to buy and why. The short email script included enough information to convey the expectations and motivations from real clients and hint the condition type. Each email script was designed so that it would remind subjects how they should process the email and to reduce the task completion time. Subjects worked with an email client that sorted customer emails and placed them in separate folders. Each of these folders had to be completed before moving to the next one. The task required subjects to scan their email folders, decide which type of customers they might be working with and what type of service to provide these customers. The subjects then found the listed price for the items requested by each customer, and arranged the products so that the customer was able to buy as many items as possible while accommodating their preferred products; all within the customer's budget. Subjects were encouraged to use their own intuition and taste in order get them involved in the task.

Two out of four emails had artificially introduced budget errors (similarly to a customer error). Errors required subjects to drop one or two items from the order, correct the item price, and recalculate the order quantities based on the customer budget restrictions. Emails with no errors received prices updates so that the update could be reflected on the email price. The errors ensured subjects devoted enough attention to the task and to keep the task from becoming monotonous. The number of items per email order varied from 3-5 items to keep the scenario more realistic and short enough not to overwhelm subjects, and to provide enough time for interruptions in the middle of the task.

Pilot Studies. Several pilot studies showed that we had indeed created a cognitively effortful scenario and that subjects should perform a minimum of two email orders before becoming familiar with the task. As a result of these pilots several changes were introduced: 1) the length of each condition was reduced to 10-15 minutes long. 2) The number of emails to be processed by condition was reduced from an earlier experiment with 6 emails to 4 emails so that subjects wouldn't be overwhelmed with too many orders needing processing. 3) The number of interruptions was reduced from 5 interruptions per email to 3 in order to allow subjects to successfully complete the order without excessive interruptions. These pilot explorations also helped choose the quality and urgency tasks to focus on in the formal experiment.

4.3 Protocol

Each mediator condition was presented in three stages, an introduction stage, a quality stage, and an urgent stage. The introductory stage served to fully familiarize subjects

with the experimental task. This within-subjects condition explored how the task is performed when the task is highly prioritized. The quality and urgent conditions were selected because they exhibited similar traits in pilot experiments. 40 subjects were randomly assigned to two conditions: Disruption Manager, and No manager. Subjects were first presented with the interface and a walkthrough of the task based on a script previously rehearsed by the experimenter. In order to obtain a consistent response to interruptions, the walkthrough included an exemplification of potential interruptions and how they should be dealt with. A practice session allowed participants to become familiar with the interface, familiar with the content, and familiar with the interrupting messages. The practice session also allowed subjects to identify the benefits from attending to interruptions. The practice session lasted until subjects completed all questions and were satisfied with their answers. On a second practice run, timed sections were introduced in order to introduce this feature and allow subjects to experiment with different navigation techniques.

4.4 Results

The hypothesis regarding performance was confirmed. Mediating interruptions yielded higher performance than without mediation supported by planned comparisons indicating a significant difference on performance based on the manager type $F(1, 37) = 473.92$, $p < .001$. The disruption manager conditions showed 26% and 32.5% performance increase over no mediation for the quality and urgent tasks respectively. Additionally to the increases in individual performance, Disruption Manager subjects were more able to share pricing information by replying to IMs, therefore improving overall goal completion or company's profits. This was confirmed by the hypothesis stating that subjects under the Disruption Manager condition would be more efficient in their task. The ratio of IMs responded to was about 5 times higher for the disruption manager condition. This indicated that the manager did much better at presenting interrupting messages (relevant information) at the right time. Participants on the manager condition responded to 58% and 51% of the IMs received for quality and urgent tasks. Whereas, they only responded to 12% and 8% of the messages on the No-manager condition, see **Fig. 4**.



Fig. 4. Instant Message Response Ratio: Subjects responded to instant messages 5 times more when using Disruption Manager while dealing with quality and urgent goals

Perceived Disruption. There was no main effect of manager type in perceived disruption $F(1,37) = .089$ $p=.7$, nor were there any significant contrasts between Quality and Urgent tasks. Thus, our third hypothesis that subjects would perceive less disruption was not confirmed. Both manager categories demonstrated a similar moderate (0.5) disruptive effect across all task categories: 0.5 - 0.6 with no manager and 0.5 – 0.5 with disruption manager. While more data might show a main effect here, the shear point that automated disruption management did not increase perceived disruption and that Subjects were not aware of the system being different is clearly exciting and present here.

5 Discussion

We have presented a Disruption Management Framework and Disruption Manager System that reduces disruption by taking into account the user goals tasks, and activities. Our results show a general computer interface capable of evaluating incoming interruptions in relation to their benefits to the user's goals and the disruption to the ongoing task, improves performance and overall productivity. Our evaluation shows a 26% and 32.5 % performance increase for task prioritized by quality and urgency. The manager also increased goal and task efficiency by presenting relevant information at the right time and allowing participants to respond to 5 times as many interrupting messages. Furthermore, participants showed no perceived difference in disruption for IM's delayed by our intelligent system. Delaying IM's by variable amounts of time did not interfere with the user goals and did not increase perceived disruption. Disruption manager can in fact reduce unnecessary disruptions without reducing a person perceived control. This work represents a new style of interface, one that can recognize and mediate important communication effectively. The approach improves a user's ability to work fast or accurately by attending to what the computer knows will be more relevant.

6 Conclusions and Future Work

In an increasingly complex information world, mediating information is critical and new tools must be developed to gather information about attention from low-level data. However, we must consider that as new sensors are being developed, their implementation is time consuming and user acceptance increases slowly. Therefore, domain-independent sources of information are still usable as means to provide fail-safe disruption mediation. In this paper we used several tools developed to investigate attention and user interest from domain in dependent low-level metrics. These tools demonstrate that a disruption manager is able to rely on Goals, Tasks, Implicit Metrics, Activity, Mediator Filters, Sensors and virtual sensors. These tools are bound to improve; everyday new technologies and approaches provide the much-needed insight into people's computing activities.

More and more software acquires an ability to decide when it is appropriate to accept or project information to and/or from particular people, particular places, and particular projects. This will have profound influence on the social dynamics between people, the ability for people to accomplish their work, their homework, their social responsibilities, and even their own goals. In addition to mediating interruptions based

on goal-related information, it is important to understand how people manage interruptions and what makes interruptions be perceived as disruptive. Future work should investigate the factors that influence people's decision process at turning interruptions into actual disruptions. For instance, interruptions relevant to user's goals might not be perceived as disruptive, even if unrelated to the ongoing task. In these situations people might be more likely to switch to the interrupting task in order to accomplish their goals, or not. Further information is needed regarding the user task priorities and their influence on perceived disruption: How do changes in task priority affect people's reactions to interruptions? What are the people's strategies and reactions for dealing with interruptions?

References

1. Altmann, E.M., Gray, W.D.: Managing attention by preparing to forget. In: Proceedings of the IEA 2000/HFES 2000 Congress, pp. 152–155. Human Factors and Ergonomics Society, Santa Monica (2000)
2. Arroyo, E., Selker, T., Stouffs, A.: Interruptions as Multimodal Outputs: Which are the Less Disruptive? In: IEEE International Conference on Multimodal Interfaces (ICMI 2002), Pittsburgh, PA, pp. 479–483 (October 2002)
3. Arroyo, E., Selker, T.: Self-adaptive multimodal-interruption interfaces. In: Proceedings International Conference on Intelligent User Interfaces IUI 2003 (2003)
4. Bailey, B.P., Konstan, J.A., Carlis, J.V.: The effect of interruptions on task performance, Annoyance, and Anxiety in the User Interface. In: IEEE International Conference on Systems, Man, and Cybernetics (2000a)
5. Bailey, B.P., Iqbal, S.T.: Understanding Changes in Mental Workload During Execution of Goal-directed Tasks and Its Application for Interruption Management. *ACM Transactions on Computer Human Interaction (TOCHI)* 14(4), 21–56
6. Bureau of Labor Statistics Current Population Survey for Computer and Internet Use at Work, <http://www.bls.gov/news.release/ciuaw.nr0.htm>
7. Burmistrov, I., Leonova, A.: Do interrupted users work faster or slower? The micro-analysis of computerized text editing task. In: To appear in: Proceedings of 10th International Conference on Human-Computer Interaction (HCI International 2003) (2003)
8. Chen, M., Anderson, J.R., Sohn, M.: What Can a Mouse Cursor Tell Us More? In: Correlation of Eye/mouse Movements on Web Browsing. In: Ext. Abstracts CHI 200. ACM Press, New York (2001)
9. Claypool, M., Le, P., Waseda, M., Brown, D.: Implicit interest indicators. In: Proceedings of the 6'4 International Conference on Intelligent User Interfaces (IUI 2001), USA, pp. 33–40 (2001); Covey: pp. 150–152 (1989)
10. Covey, S.R.: *The Seven Habits of Highly Effective People*. Simon and Schuster, Inc., New York (1989)
11. Cutrell, E., Czerwinski, M., Horvitz, E.: Notification, Disruption and Memory: Effects of Messaging Interruptions on Memory and Performance. In: Hirose, M. (ed.) *Human-Computer Interaction INTERACT 2001*, Tokyo, July 9–13, pp. 263–269. IOS Press(for IFIP), Amsterdam (2001)
12. Czerwinski, M., Cutrell, E., Horvitz, E.: Instant Messaging and Interruption: Influence of Task Type on Performance. In: Paris, C., Ozkan, N., Howard, S., Lu, S. (eds.) *OZCHI 2000 Conference Proceedings*, pp. 356–361 (2000-B)

13. Czerwinski, M., Cutrell, E., Horvitz, E.: Instant Messaging: Effects of Relevance and Time. In: Turner, S., Turner, P. (eds.) *People and Computers XIV: Proceedings of HCI 2000*, vol. 2, pp. 71–76. British Computer Society (2000)
14. Czerwinski, M., Horvitz, E., Wilhite, S.: A diary study of task switching and interruptions. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Vienna, Austria, pp. 175–182 (2004)
15. Gillie, T., Broadbent, D.: What makes Interruptions Disruptive? A study of length, Similarity and Complexity. *Psychological Research* 50, 43–250 (1989)
16. Gievska, S., Sibert, J.: Examining the Qualitative Gains of Mediating Interruptions during HCI. In: *The Proc. of HCI 2005* (2005)
17. Groff, B.D., Baron, R.S., Moore, D.L.: Distraction, attentional conflict, and driveline behavior. *Journal of Experimental School Psychology* 19, 359–380 (1983)
18. Hess, S.M., Detweiler, M.C.: Training to reduce the disruptive effects on interruptions. In: *Proceedings on the Human Factors and Ergonomics Society 38th Annual Meeting*, pp. 1173–1177 (1994)
19. Horvitz, E., Apacible, J.: Learning and reasoning about interruption. In: *Proceedings of the Fifth International Conference on Multimodal Interfaces*, Vancouver, BC, Canada, vol. 29 (November 2003)
20. Horvitz, E., Kadie, C., Paek, T., Hovel, D.: Models of attention in computing and communication: From principles to applications. *Communications of ACM* 46(3), 52–59 (2003)
21. Iqbal, S.T., Horvitz, E.: Disruption and Recovery of Computing Tasks: Field Study, Analysis and Directions. In: *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2007)*, San Jose, California, USA, pp. 677–686 (2007)
22. Iqbal, S.T., Bailey, B.P.: Understanding and Developing Models for Detecting and Differentiating Breakpoints during Interactive Tasks. In: *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2007)*, San Jose, California, USA, pp. 697–706 (2007)
23. Jackson, T.W., Dawson, R.J., Wilson, D.: The cost of email interruption. *Journal of Systems and Information Technology* 5(1), 81–92 (2001)
24. Latorella, K.: Effects of Modality on Interrupted Flight Deck Performance: Implications for Data Link. In: *Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting*, Chicago, IL (October 1998)
25. Lieberman, H., Selker, T.: Out of Context: Computer Systems that adapt to, and learn from, context. *IBM Systems Journal* 39(3-4), 617–632 (2000)
26. Locke, E.A., Latham, G.P.: Building a Practically Useful Theory of Goal Setting and Task Motivation: A 35-year Odyssey. *American Psychologist* 57, 705–717 (2002)
27. Maes, P.: Agents that Reduce Work and Information Overload. *Communications of the ACM* 37(7), 31–40 (1994)
28. Scott, M.D., Chewar, C.M.: Attuning Notification Design to User Goals and Attention Costs. *Communications of the ACM* 46(3), 67–72 (2003)
29. McFarlane, D.C.: Coordinating the interruption of people in human-computer interaction. In: Sasse, A., Johnson, C. (eds.) *Proceedings of Human-Computer Interaction (INTERACT 1999)*. IFIP, pp. 295–303, page 121. IOS Press, Amsterdam (1999)
30. McFarlane, D.C., Latorella, K.A.: The scope and importance of human interruption in human-computer interaction design. *Human-Computer Interaction* 17(1), 1–61 (2002)
31. Nagel, K.S., Hudson, J.M., Abowd, G.D.: Predictors of availability in home life context-mediated communication. In: *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*, Chicago, Illinois, USA, November 06-10 (2004)

32. Rasmussen, J.: Information processing and human-machine interaction. An approach to cognitive engineering, p. 215. North-Holland, New York (1986)
33. Singh, P., Lin, T., Mueller, E.T., Lim, G., Perkins, T., Zhu, W.L.: Open Mind Common Sense: Knowledge acquisition from the general public. In: Chung, S., et al. (eds.) CoopIS 2002, DOA 2002, and ODBASE 2002. LNCS, vol. 2519. Springer, Heidelberg (2002)
34. SuwatanaPongched, P.: A More Complex Model of Relevancy in Interruptions (2003)
35. Tsukada, K., Okada, K.I., Matsushita, Y.: The Multi-Project Support System Based on Multiplicity of Task. In: Eighteenth Annual International Computer Software and Applications Conference (COMPSAC 1994), pp. 358–363. Institute of Electrical and Electronics (IEEE), New York (1994)
36. Van Bergen, A.: Task interruption. North-Holland, Amsterdam (1968)