NORTHWESTERN UNIVERSITY

Heed: A Framework For Situation Aware Monitoring

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILIOSOPHY

Field of Computer Science

By

Conrad Albrecht-Buehler

EVANSTON, ILLINOIS

December 2008

Abstract

Heed: A Framework For Situation Aware Monitoring

Conrad Albrecht-Buehler

Technology users are technology observers as well, monitoring for problems or opportunities that might arise. Designing interfaces to support the monitoring of technology presents unique challenges for everything from detecting situations to responding to situations to changing the tools to match a changing operating environment or the observer's changing knowledge. An interface needed to overcome these challenges, and the process of designing such an interface must take all of these challenges into consideration. I present "heed": a scale and framework to help observers of a system evaluate which situations need scrutiny and when, along with an example heed-based interface that encourages the development of situation awareness. By presenting the importance of attending to each situation on a scale instead of in a binary state, observers are able to classify which situations need their attention and which can be safely ignored, and by presenting how that importance is changing, they can estimate if a situation might need their attention soon. The heed framework makes it possible for observers to describe situations in terms of a rich set of conditions and logical operators, and the interface enables them to refine their descriptions and perceive the importance of attending each situation. I discuss how the framework and interface can be applied to the monitoring of four very different situations: server performance, a business's finances, user experience in a community forum, and the risk of disease outbreak at a veterinary clinic, and how these examples can be generalized to guide the application of heed to many other situations.

Acknowledgements

No words can express my gratitude to the many people who made this work possible. I am deeply grateful to my advisor, Don Norman, who influenced my thinking long before I ever met him, and inspired me every day thereafter. I thank him for always challenging me, listening to me, being patient with me, and his many years of teaching me. He taught me to question conventional wisdom and to strive to empower people to do what they do best. I am grateful to my committee Andrew Ortony and Bill Revelle for their council, wisdom, and insight. It was an ineffable honor to have worked with them.

My many thanks to Carl Albrecht-Buehler, Dan Ratner, and Shinji Kuwayama. Through all their wonderful support and insight, they became my ad-hoc labmates. They championed this work, kept me going, and helped to make every idea a reality; I could not have done this without them.

A very special thank you to Ben Watson, who shepherded me through the difficult times of being a young graduate student and taught me how to make tea.

I wish to acknowledge the many people in the computer science department who've made my time there so memorable. Chris Riesbeck, who taught me how to teach and always helped me move forward, and Kris Hammond, Bryan Pardo, and Larry Birnbaum who taught me by example. So many wonderful graduate students made my time there special, but I wish to thank Kevin Livingston and Brooke Foucault for their advice on this work, and Ayman Shamma for your many years of friendship and for teaching me how to draw.

Thank you to Hannah, Stephanie, Carolyn, and all the gang at Julius Meinl for making me feel welcome and giving me a place to write.

My deepest gratitude to my close friends who held me up when it was hard to stand: Dan, Genevieve, Shinji, Stacy, Josh, Steve, Jim, Kristin, Justine, and Kate. I don't know what I did to deserve such wonderful friends.

It is to my family that I owe the greatest thanks. My mother, Gabriele, who exhausted every avenue to help me any way that she could. My father, Guenter, who challenged my every assumption. My brother, Carl, who helped me see what was right in front of me, and his wife Heather, who laughed at me whenever I took myself too seriously. My sister, Christine, who is always looking out for me, and her husband Josh, who taught me how to keep perspective. And my nephews, Huck and Hugo. I can't wait to see what wonderful things you'll do one day.

Finally, I will forever be grateful to Jennifer for her tireless support and boundless love. When times were bad, she made them better; when times were good, she made them possible.

Thesis Committee

Donald A. Norman, Northwestern University, Committee Chair

Andrew Ortony, Northwestern University

William Revelle, Northwestern University

To Jennifer

Table Of Contents

List of Figures

Chapter 1: Introduction

Problem

Technology users are technology observers as well. Users interact with technology to perform a task, and observers monitor the technology for problems or opportunities that might arise. The distinction is not important for simple systems, but the more complex the system, the more these roles differ. Designing interfaces to support the monitoring of technology presents unique challenges for everything from detecting situations to responding to situations to changing the tools to match a changing operating environment or the observer's changing knowledge. An interface needed to overcome these challenges, and the process of designing such an interface must take all of these challenges into consideration.

Let's consider a hypothetical situation: Maria is a systems administrator for a small company. Her job requires her to be aware of the status of the various servers used by the company, to perform upgrades to the servers, to correspond with users and managers, to purchase new equipment, to perform basic repairs on existing equipment, and to document her work,. Because she has so many responsibilities, Maria maintains her awareness of the servers in three ways, each intended to give her more time to perform her other duties. One: she relies on alarms she receives as pager notifications or phone calls from upset users to tell her when something has reached critical status or has failed. Two: she skims a dashboard of graphs of system parameters looking for something to "jump out" at her as "unusual." Three: she looks over reports generated by a system analysis tool to tell her how particular aspects of the system have been performing. Unfortunately, each method she employs falls short in some way, and as result, problems arise in the system seemingly without warning. Alarms and user calls only tell her that something has already happened, and the point of her keeping tabs on the servers is to intercede before a problem arises. Furthermore, an alarm or user call could come at any time, which interferes with decisions as simple as when to go to lunch. Because she always has to choose to either drop what she's doing and address the problem or to ignore the problem and let it continue, she feels like she's "always putting out fires." If she can't respond to an alarm right away, what else can she do about it? Let the phone keep ringing? Let the alarm continue to sound? If she cancels the alarm, will she remember to look into the situation later? Skimming her dashboard of graphs has its own shortcomings. Although it might enable her to identify a problem before it occurs, it's quite time-consuming to do, and she often prioritizes her other duties over this monitoring activity. In addition, this method relies on her ability to recognize something unusual in the various displays. The less she looks at them, the harder it is to know what's usual and what's not. Furthermore, a situation might be indicated by a combination of features from several graphs - what if she doesn't look at the "right" ones, or if she forgets what's in one while skimming over another? Finally, her reports might do a great job of analyzing the system for situations she cares about, but if her system is relatively stable, there's usually nothing of interest to note. At best, then, this skimming is a waste of her time, and at worst, she becomes complacent and stops doing it altogether.

A better interface would enable Maria to determine when the reports are worth looking at, or when it might be a good use of her time to skim the dashboard, but it should also, if she can't drop what she's doing, continue to tell her that there is a situation that needs her attention. If possible, it should also enable her to make tactical decisions by telling her that a situation might need her attention in the near future, even if it does not yet. In other words, it should tell her which situations she should take heed of and when.

Introduction

To address these design criteria, I propose the use of a scale to classify the importance of attending to a situation, which I call “heed.” The scale extends from “safe to ignore” at 0 to “must attend” at 1. The scale is intended to be indifferent to the type of situation and its potential outcome, which makes it possible to compare situations regardless of their subject. Thus, Maria can decide that situation A, a disk cache that is more than half full, at 0.3 heed probably isn’t critical enough for her to drop what she’s doing; namely, investigating situation B, an unusual amount of email being sent, which is at 0.8 heed. Furthermore, the situation classification can be extended by including first and second derivatives of heed. Thus situation A could be described as at 0.3 heed and increasing at the rate of 0.05 heed/minute.

All of Maria's existing monitoring tools are based on sensor readings, from the alarm that triggers when the CPU temperature reaches 160°F to the dashboard that shows her bandwidth usage and the report that shows her email activity. The heed value of a situation is based on these sensor readings. Each sensor reports values using different units of measurement, but the heed scale is intended to be a universal basis into which all sensor units can be transformed. Transforming sensor readings into heed values relies on the knowledge of the observer or another system expert, and is accomplished by a rule arithmetic.

To illustrate how heed could be applied, consider a simple situation that hinges on only one sensor reading: is the disk full? At minimum, this situation only needs to know the remaining capacity of a disk. A more complete view would take into account how quickly the disk is filling up and how much the running applications will contribute to that fill rate, but for this example, I’ll ignore those factors (I return to them in chapter 5).

Suppose there is a 300 gigabyte (GB) disk. The heed value requires someone (perhaps Maria herself) to determine at what point it is important to pay attention to the amount of free space left on the disk. In this case, assume that from her previous experience with the system, Maria has determined that 10 GB remaining is the critical point at which attention should be paid to the system. Heed is a function based upon an assessment of the criticality of the remaining disk space, with a minimum value (when there is sufficient disk space not to need any concern), a maximum (when no more disk space is available), and a transition to increasing heed based upon the critical value (which, in this example, is 10 GB).

Outline

In Chapter 2 I discuss how the complexities of monitoring tasks have been addressed in the fields of situational awareness and supervisory control. Next, I discuss work related to alarms and other means of informing system observers of situations when they arise. Finally, I discuss prior work that has examined how people learn about complex systems.

Chapter 3 discusses a procedure for calculating heed and chapter 4 discusses an example interface for communicating and interacting with heed. Chapter 5 includes a discussion of four case studies in which I apply the heed scale to very different situations. I use these case studies to illustrate the variety of ways in which the methodology and example interface can be applied, how subjects responded to the interface, and how to create similar heed-based applications. I conclude in chapter 6 with a discussion of where improvement is needed in the interface and how this work might be applied and extended in the future.

Chapter 2: Previous and Related Work

Introduction

Monitoring does not happen for its own sake. It serves a purpose, and the models of interaction that I discuss at the start of this chapter all treat monitoring as an aspect of maintaining control over a system or a process. I first discuss some of the most influential models of control that include monitoring, namely supervisory control (and work corollary to supervisory control) and situation awareness, and highlight the role that monitoring plays in these models. Then I discuss methods and interfaces for monitoring, and focus on those that support reacting to events and those that support predicting events. Finally, I discuss work that has explored how people learn about complex systems.

Sheridan

One of the earliest and probably most cited models of human functioning in complex environments is Sheridan's model of supervisory control (Sheridan 1976). The model describes the relationship of a human operator to a process or task wherein a computer acts a mediator. In a supervisory control mode, the human loosely exerts control over the task by guiding the computer which is in direct control of the task. Specifically, Sheridan defines supervisory control as "…one or more human operators [which] are continually programming and receiving information from a computer that interconnects through artificial effectors and sensors to the controlled process or task environment." He describes the sort of programming the human operator performs in a supervisory control modes as "…specifying to the computer goals, objective tradeoffs, physical constraints, models, plans, and "if-then-else" procedures."

Sheridan defines five functions for the human supervisor in the supervisory control mode as follows:

1) to plan what task to do and how to do it
2) to teach the computer what that plan is
3) to monitor the computer's progress toward fulfilling the plan, and to detect failure
4) intervening when progress is faulty or ineffective
5) learning from the experience to improve future activity (Thomas 1992)

Sheridan presents these rules as a simple set of control loops that describe the time-order of how a task can be completed in a supervisory control system. The control loops are the fundamental model of the supervisory control system. The third rule defines the specific role of monitoring in a supervisory control system: to gauge progress and detect failure. Furthermore, he goes on to assert a vital requirement of supervisory control: "A very important aspect of supervisory control is the ability of the computer to "package" information for visual display to the human operator. Data may be included from many sources, from the past, the present, or even the predicted future, presented in words, graphs, symbols, pictures, or some combination." (Thomas 1992) What Sheridan highlights here is that a process without direct control requires the ability to make observations.

The model, and particularly Sheridan's five functions, highlights the necessity for the human to be informed about the state and progress of the computer's functioning, and that, in a complex environment, there is continual opportunity to learn and improve. Many supervisory control systems have overlooked the importance of supporting learning and improvement.

Rasmussen

Rasmussen (1986) expanded Sheridan's description of human functioning in supervisory control modes by classifying the human's behavior. He proposed a three-level model of increasing cognitive involvement: the so-called "Skills, Rules, Knowledge" model. This model takes the learning aspect of interacting with the complex environment even further by accounting for why some situations are easier to handle than others.

Once an operator has chosen a desired outcome (i.e., formed an intention), skill-based behaviors require little conscious involvement to execute and occur smoothly and require few cognitive resources. Rule-based behaviors require more cognitive resources, and are characterized by executing a series of rules or procedures in response to a familiar situation. Finally, knowledge-based behaviors involve significant cognitive resources as the operator copes with an unfamiliar situation. At this level, behavior is governed by the degree of training and expertise the operator has as well as the quality and appropriateness of the information the operator has access to.

Behavior at each level is dependent on the ability of the operator to recognize the situation, and that recognition is limited by the quality, timeliness, and appropriateness of the information available to the operator. The information, when appropriate, can trigger skill-based behavior, but if inappropriate, it forces knowledge-based behavior in order to investigate the situation. Furthermore, the degree of operator expertise influences situation recognition. In fact, expertise and training, over time, can change knowledge-based behaviors into skill-based ones, but this can only occur through a process of learning. Learning and recognition are both dependent on the information an operator receives and therefore are both dependent of the monitoring tools available to the operator.

Joint Cognitive Systems

Hollnagel and Woods (2005) proposed an expansion of the classical supervisory control model by advocating a greater degree of cooperation between the computer and human. The crux of their model, which they call a Joint-Cognitive System (JCS), is the tight cooperation of the human and machine, not just for task control, but also for analysis and investigation. One way to think about the JCS model is to think of the original Sheridan model, but instead of the human alone monitoring, programming, and intervening, it is a human working with a computer to do so. This model accounts for the fact that many supervisory control applications are so complicated that the activities Sheridan's model expects the human to do are too complex for that human to accomplish without assistance.

The authors propose a guideline which they call "designing for complexity" as being the best way for a JCS to cope with a complex system. They describe the concept as follows: "Designing for complexity fully acknowledges that the work environment of the operator is complex but draws the conclusion that this complexity is necessary for effective control. ...Designing for complexity aims to support general functions of coping, rather than specific ways of acting in particular situations, hence chooses generality over specificity." To accomplish this, the authors describe two principles that designers should employ. The first is to support the natural strategies of coping that an operator is familiar with and would normally use instead of trying to enforce a particular strategy. The notion here is essentially a user-centered one that advocates studying the operator activity and providing support of those activities instead of making assumptions about the operator's state of mind. Interestingly, the authors make no mention of enabling the operators to create or modify the interface themselves. The next principle they advocate is providing the JCS with sufficient time to respond, and they posit that one way of accomplishing this is through situation forecasting.

Signal Detection Model

Where Hollnagel and Woods proposed a high-level cooperative environment in which a human and automated system operate together to perform complex tasks, Sorkin and Woods propose a low-level cooperative model wherein a human and automated system perform the monitoring task together. The Sorkin and Woods model is the basis for describing the role of alarms in monitoring.

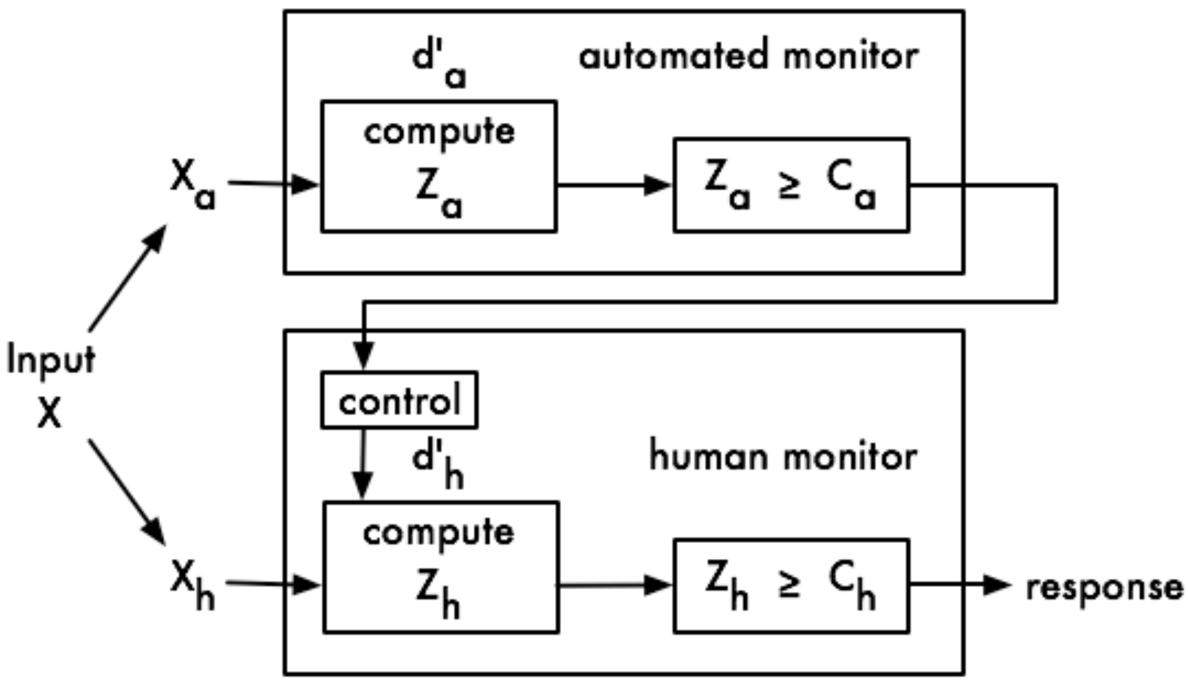


Figure 1: Sorkin and Woods signal detection model (Sorkin and Woods 1985)

The Sorkin and Woods model is based on basic signal detection theory, and treats the combined human operator and automated monitor subsystem as a dual-stage signal detection system (Sorkin and Woods 1985). The model is as follows: the human and the automated component monitor partially correlated noisy channels of an input signal. Each processes the channel input and calculates a measure of signal strength in the original input, and that value is compared to a decision threshold. If the value is greater than or equal to the decision threshold, the automated subsystem sends an alarm to the human operator. If the human has not processed the input channel yet, the alarm signal initiates that behavior; namely it signals that the operator should begin monitoring. The human is modeled as performing a similar activity to the automated system that results in comparing a signal strength with a decision threshold and based on that comparison, initiating a response. In this model, the operator could monitor the input channel without being initiated by the alarm. The alarm signal does not require monitoring by a person.

Sorkin and Woods treat monitoring of the signal as separate partially correlated channels because the human and automation systems have different knowledge and decision criteria. The human may have unique access to historical or contextual information and the automated system may have been knowledge the operator either does not know or is not capable of computing. The channels are treated as noisy because the ability to sense the original signal can be hampered by inaccurate sensors or imperfect sampling.

Sorkin and Woods conclude that it is important to design the automated monitor based on the complete human-automated monitor complex. The choice of decision criterion has a large impact on the performance of the entire system. They found that if the decision threshold is chosen to minimize the number of misses, the total performance diminishes, akin to an overly sensitive alarm that is too easily triggered and real-world observation. See the discussion of the effects of false alarms in (Breznitz 1984), (Bliss and Gilson 1998), (Bliss, Gilson et al. 1995), and (Lehto 1998), as well as (Parasuraman and Riley 1997).

C-HIP Model of Warning

Wogalter et al. (1999) describe a model of the warning process based on communication theory and information processing theory. They refer to the model as a Communication-Human Information Processing (C-HIP) model, and it is a combination of the two theoretical frameworks. The communication theory model of Source, Channel, Receiver is modified by detailing the receiver stage with the information processing theoretical model of attention, comprehension/memory, attitudes/beliefs, motivation, and behavior.

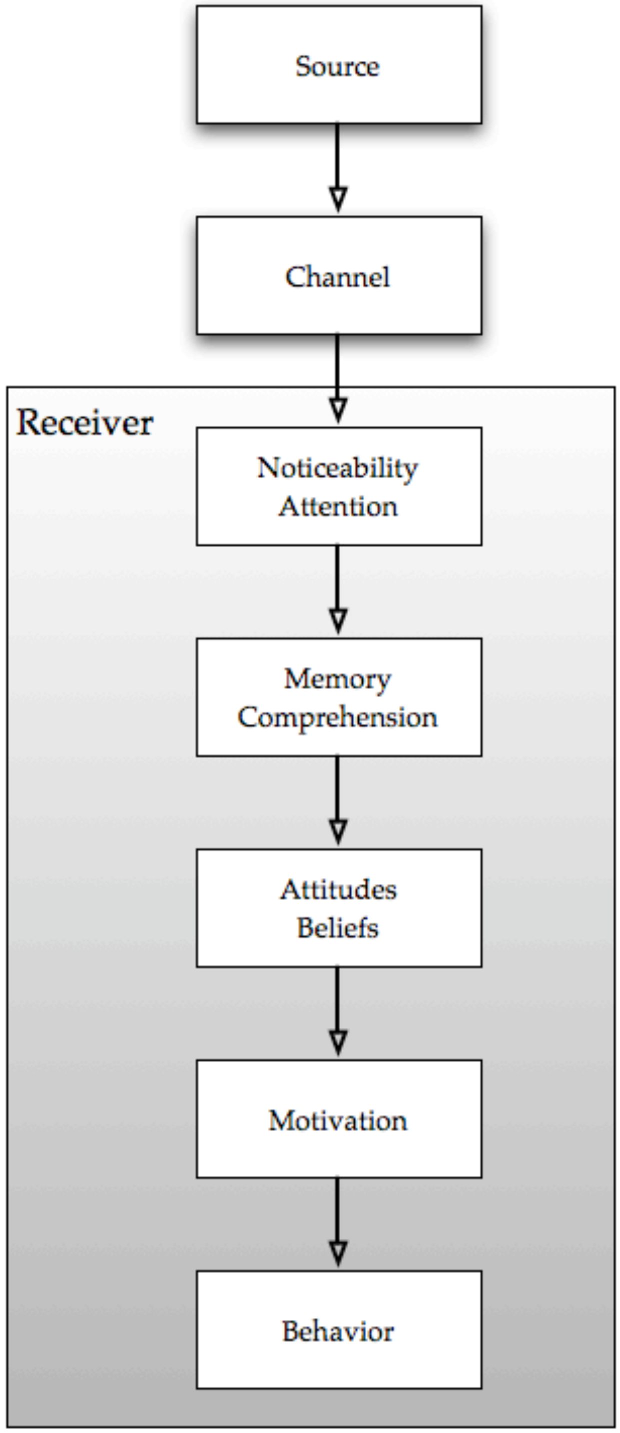


Figure 2: The Communication—Human Information Processing model of warning processing (Wogalter, Dejoy et al. 1999)

Building on this model, Young and Lovvell (1999) point out that attention must be paid to the design of warnings (or tools that warn) to account for the intermediate stages of information processing if the warning is to be effective. They note that most research on warnings focuses on attention and behavior, but often overlooks the intermediate stages. They highlight each stage and make recommendations for how to test and improve each one. Of interest are their recommendations for comprehension, memory, attitudes and beliefs. With regard to comprehension, they conclude that the verbal and pictorial aspects should be treated as a unit, not separately. The verbal can convey more precise information, but takes longer to process. They also point out that using warnings authored by the person being warned is a powerful way of increasing comprehension. With regard to memory, they discuss how to distinguish guessers from those that actually recall the warning itself. Here they highlight the importance of labels as a way to improve recall. Finally, the authors discuss how attitudes and beliefs affect willingness to comply with warnings as well as the interpretation of warnings. They note that not much research has been done in this area, and that understanding the population that is expected to heed the warning is important to improving compliance and understanding. They do not, however, mention how the processing of self-authored warnings is influenced by attitudes and beliefs.

Situation Awareness

In Sheridan's model of supervisory control, the third function that the human performs is that of monitoring progress and detecting failure. Situation awareness (SA) is defined by Endsley as "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future" (Endsley 1988). "Basically, SA is being aware of what is happening around you and understanding what that information means to you now and in the future. This awareness is usually defined in terms of what information is important for a particular job or goal." (Endsley, Bolté et al. 2003). In other words, SA is a mental model constructed by an individual, specific to the situation and goal.

To establish SA, an individual must continuously develop and refine a mental model of all the conditions that influence the situation. Endsley and her coauthors describe the development of SA in the individual as occurring at three levels, each resulting in a more complete mental model of the operational environment and the many factors influencing progress toward a goal: perception, comprehension, and projection. The first level deals with perceiving "… the status, attributes, and dynamics of relevant elements in the environment" (Endsley, Bolté et al. 2003) and comes directly from the senses. She points out that "[i]n many complex systems, a strong emphasis is placed on the electronic displays and read-outs that are provided, but the reality is that much of Level 1 SA also comes from the individual directly perceiving the environment — looking out the window or feeling the vibration." (Endsley, Bolté et al. 2003) Perception of relevant elements is hindered by many factors: failing to provide information to the person who needs it, not providing it clearly, forgetting what information was detected, or being distracted and unavailable to detect important information.

The second level of SA is that of comprehension and involves assimilating the data from level 1 with the goals of a task, a knowledge base of experience, and a mental model of the operational environment. The establishment of level 2 SA is based on acquiring the appropriate information, parsing and filtering it, and associating meaning with it.

Finally, the third level of SA occurs when the person is able to forecast what the elements relevant to a goal will do so as to make proactive decisions. The ultimate goal of supporting SA is to support the establishment of level 3 SA within the individual. In order to establish level 3 SA, the individual needs sufficient mental resources not otherwise occupied, sufficient domain knowledge to understand how the operating conditions could change, and finally, not to over-project current trends. Any tools intended to support the monitoring of a complex system must support the establishment of all three levels of SA.

Tools That Support Reacting To Events

An alarm is a simple and effective means for overcoming the human limits of sense and attention. However, the misuse of alarms can lead to a great loss of control. Several authors have explored the role of alarms and how their inclusion in a monitoring interface impacts user activity.

Alarms and Being Alarmed

The Sorkin and Woods model of signal detection describes how a monitored alarm system ought to work, and when an alarm might be triggered, but an important aspect of how the human in the model should respond to an alarm is often ignored. Their model suggests that when an alarm is triggered, the human first confirms the signal and then initiates a response to it. Many authors have discussed what happens when operators and interface designers ignore that first step.

Stanton points out that an alarm is often badly defined as "… a significant attractor of attention" or "a piece of information"(Stanton 1994). He notes that "the term may be used to define both the stimulus and the response on different occasions. In a stimulus-based model an alarm exists in the environment and its presence has some effect on the individual, whereas in a response-based model, the stimulus causes an alarm state in the individual." This destination is important for alarm and monitoring systems designers. Is their goal in having the monitoring interface trigger an alarm to initiate a particular response, essentially saying:

“now is the right time to perform Task A,” or is the goal to initiate increased vigilance on the part of the operator, otherwise phrased as “Now is a valuable time for you to be concerned about Task A”? The problem lies in the assumption that all people will respond the same way to the same alarm. Attending an alarm can have a negative payoff; that if too much time is spent responding to alarms that don’t result in a confirmation (that is, to false alarms), important alarms will begin to be ignored (Breznitz 1984).

The Management of Alarm Systems

A 1998 study conducted by Bransby and Jenkinson for the U.K.’s Health and Safety Executive studied how alarms were used and deployed at petrochemical plants (Banbury, Macken et al. 2001). As part of their study they outlined several methods for assessing the performance of an alarm system, based mostly in the practical impact of the alarm on operations. They assessed the performance of an alarm system by counting the number of alarms during a major plant upset and the number of standing alarms during typical operating conditions, as well as just a quick determination of an average alarm rate. They also used operator questionnaires to examine alarm usefulness and to identify frequent or insufficient alarms. However, it is their recommendations for improving an alarm system based on the performance that are of particular interest, since these recommendations can be read as design guidelines for a monitored alarm system such as the one discussed in this dissertation.

The authors advocate the establishment of a culture of improvement within the plant wherein individuals or select teams are tasked with improving alarms regularly through a process of alarm review and supported by a policy of no-blame reporting, operator involvement. An alarm review can focus on identifying nuisance alarms, can be more thorough by examining each alarm starting with highest priority ones, or can be incident-specific and examine all those alarms identified during a Hazard and Operability study.

By advocating a process of regular alarm review the authors acknowledge that an alarm system is not a static operational entity and that optimum performance can only be maintained through frequent and regular modification of the alarm system. The authors discuss many techniques for improving an alarm system following an alarm review, including techniques for limit modification, alarm re-definition and re-engineering, alternate alarm presentations, alarm suppression, and addressing repeating and fleeting alarms.

Their recommended techniques for modifying limits include tuning the limits of known nuisance alarms to better capture the hazard the alarm was designed to detect, changing the alarm to trigger based on deviation instead of absolute value, or introducing a system to automatically adjust the limits based on a predefined logic. They also suggest that another approach, though not yet frequently applied, could enable the operator to manually adjust the limits to cope with nuisance alarms, or in fact, give that operator full control over all alarm parameters. The notion of frequently adjusting alarm limits as well as that of putting the control of those limits into the hands of the people meant to respond to the alarms does not seem to be explored anywhere else, despite the now ten-year old recommendation to do so.

Bransby and Jenkinson describe techniques for re-defining and re-engineering alarms to improve alarm performance. What the authors mean by alarm re-definition is a more strict distinction between status information and alarms. They point out that many serial alarm displays will include status changes intermingled with alarm notifications. By more strictly defining what an alarm is and where it can be displayed, alarms will become easier to find and distinguish. What the authors mean by re-engineering alarms is modifying the low-level sensing and signaling hardware. They advocate analog measurement over binary condition sensor — in their case, so that other alarm filter techniques, such as the use of a deadband (i.e. a temporary alarm limit adjustment in response to an alarm being triggered), can be applied. They advocate the use of automatic trips in place of operator alarms for extremely hazardous conditions, though they are conscious of the potential negative impact of an added automatic system. As a final re-engineering recommendation, they suggest that discrepancy alarms (those that trigger when an actuator is slow or fails to respond) should be made robust by making the alarms more aware of the actuator's degradation, or by segregating discrepancy alarms from all other categories of alarm.

The authors also mention a few techniques to improve alarm presentation. They suggest presenting alarms of different priorities differently. They also suggest adding a classification to the alarms based on which part of the plant they are triggered in. Furthermore, they would like to see related alarms grouped together; that is, they would like to see several related alarms presented as a single super-alarm representing the group. They caution that grouping alarms in this way will require the operator to investigate the alarm before deciding what to do. And they advocate clear, meaningful, and consistent alarm messages.

Finally, the authors discuss techniques to improve the performance of repeating and fleeting alarms. Alarms that trigger frequently or those that trigger too briefly and are easily missed are some of the most common nuisance alarms studied. To minimize the impact of these sorts of nuisance alarms on overall alarm performance, they suggest a myriad of potential sources of improvement. For one, they point out that locating alarm sensors in turbulent parts of the plant only adds noise to the sensor signal, thereby reducing the alarm's reliability. If an alarm sensor cannot be relocated, they suggest applying a low-pass filter to the signal to reduce noise. Several of the authors' suggestions involve processing the input signal based on situational or temporal factors, such as performing transient suppression of alarms during predictable periods in which the alarm always triggers (e.g. during a startupsequence). Among their recommendations is also what they term the logical suppression of alarms. By using additional sensor information, such as timers or plant running logic, and combing them with the alarm sensor using logic, some plants have been able to reduce nuisance alarms. Interestingly, the authors found that such complex alarms were treated as “process verifying,” and that "...many users have some resistance to any form of logical suppression of alarms (preferring an additional nuisance alarm to an incorrectly suppressed alarm)."

Other Alarm Presentations

Alarms have not just taken the form of bells, whistles, or lights. A few noteworthy alternatives are worth discussing. Stanton and Stammers (1998) discussed three general classes of alarm displays and the types of alarm handling tasks to which each is best suited. Serial, also known as chronological, are text- or speech-based alarm displays that present a time-ordered list of triggered alarms. The authors found that tasks that require time-based reasoning are best served by these types of displays. Annunciator alarm displays present a static arrangement of all possible alarms and highlight those that have been triggered. Such a display is helpful in recognizing patterns of alarms — patterns that an operator can classify as a particular familiar situation, but do not include any sort of chronological information. Finally, there are mimic alarm displays, where the alarms are displayed over a logical representation of the system operation, which puts logically related alarms near one another, making it easier to determine causal relationships.

Others have looked at modifying the alarm signal itself. Edworthy (1994) discussed how different frequencies of audio pulses as well as patterns of pulses could be mapped by individuals to different levels of urgency. Arrabito et al. (2004) looked at a broader set of auto properties, and tested combinations of high and low pitches, multiple to single tones, as well as different patterns. They also found that their test subjects could actually map different levels of urgency to each combination and that the urgency was something that they could distinguish during a simulation.

Belz et al. (1997) looked at using auditory icons, which are sounds with stereotypical meanings, like glass breaking, as the auditory warning. The authors used a method of having the users perform the mapping between the auditory icon and its perceived meaning and urgency. They then expanded the use of auditory icons with a simultaneous visual icon on a display (Belz, Robinson et al. 1999). They found that performance was significantly improved when the alarm signal combined a visual with an audio cue.

Likelihood Alarm Displays

Sorkin, Kantowitz and Kantowitz, built on the designs proposed by Sorkin and Woods and tested what they refer to as Likelihood Alarm Displays (LAD) (Sorkin, Kantowitz et al. 1988). Sorkin and Woods proposed that one way of improving an alerted-monitor system is to include additional information in the alert signal. One example, they suggested, was to include the calculated statistic used by the automated system that triggered the alert; in other words, the likelihood that a signal (in other words, an event) was detected. Sorkin, Kantowitz, and Kantowitz tested this idea by building two test interfaces. Each interface communicated the likelihood that the signal was detected precisely by an automated signal detection system. One interface communicated the likelihood with color, and the other with a synthesized voice. The authors devised a test for these interfaces by having a subject use a mouse to track a moving target drawn at the top of a computer display and monitor signal values in the form of four three-digit numbers occasionally displayed at the bottom of the computer display. The numbers represented samples from two possible normal distributions, one with a mean of 3.0 and another with a mean of 4.0, and were treated as “no-signal” and “signal” respectively. Subjects were expected to perform both the tracking and the monitoring task, and their performance in both was recorded. The test displays were enhanced with each of the likelihood displays: the tracking cursor was rendered in a different color depending on the likelihood of a simulated automated system that also monitored the signal, and would indicate to the subject the likelihood of a signal in the two renderings schemes: by changing the color of the tracking cursor in one set of tests, and by making a statement in a synthesized voice in the other. The two testing scenarios were further divided into four more, by communicating likelihood in the form of a two-state likelihood output and a four-state output, and also by changing the tracking task from an easy task to a hard task.

The authors tested with two criteria in mind: first, that performance on the monitoring task should improve over the same task without an alarm, and second, that the effect on the primary tracking task should not be significant. They found that the subjects’ tracking performance improved, accuracy improved, and reaction time decreased. In addition, they learned that the subjects’ information processing strategy changed in the presence of an automated alarm system. They concluded that when an automated alarm system was present, less attention was required for the monitoring task. In tests of two- vs. four-state displays, they found that the increased information resolution did not increase or decrease performance significantly, so they concluded that the likelihood information did not have a cognitive “overhead”. Furthermore, the type of output, in this case speech vs. visual, did not make a difference either, indicating that a separate channel for the likelihood information outside of the channel used for tracking was not necessary to improve performance.

The Impact of Alarms on SA

In their discussion of how alarms impact SA, Endsley et al. describe the impact of missed alarms and false alarms on performance in light of operator workload. Under normal, i.e. moderate, workloads false alarms are a nuisance and can increase workload, but missing an alarm doesn't necessarily have a negative impact on the task. However, during periods of overload, i.e. very high cognitive and attention loading, a missed alarm can be very detrimental to a task, but a false alarm would not be as hazardous. Of particular interest is their conclusion that during periods of underload, where long periods of time can elapse with nothing happening, not only is a missed alarm very hazardous, but, surprisingly, a false alarm can actually increase vigilance and performance. Basically, long periods without any events lead to a decrease in vigilance. Signals help maintain the vigilance level. Some studies, for example Wilkinson (1964), support the injecting of fake error signals amongst the real ones to maintain vigilance.

It isn't enough to detect an alarm: it must be diagnosed as well. Endsley et al. (2003) point out that diagnoses can be hindered in at least two ways: representational errors (settling on a single fault diagnosis when multiple faults are the cause) or through an overwhelming number of alarms that must first be filtered or parsed before beginning diagnosis. A representational error is one in which the alarm is perceived but its significance is not because the recipient associates it with an incorrect scenario. Essentially, if an alarm is triggered and the operator assumes a cause based on an erroneous mental model, the alarm will be misdiagnosed. Similarly, if the operator identifies a potential cause for a triggered alarm and responds based on the single cause when, in fact, multiple causes are at fault, the alarm is likewise misdiagnosed. Furthermore, correctly diagnosing an alarm is hindered when there are many alarms are triggered concurrently. Finally, Endsley et al. also point out that alarm reduction schemes such as filtering based on anomaly detection and classification can reduce the occurrence of misdiagnosis.

Based on their study of the impact of alarms on SA, the authors propose nine design principles intended to improve the use of alarms in support of establishing and maintaining SA. These principles were considered and used as guidelines in the design of the system presented in chapter 4. They are as follows:

1. Don't make people rely on alarms - provide projection support

They advocate the development of monitoring interfaces that support being proactive rather than relying on alarms to simply react to unexpected conditions.

2. Support alarm confirmation activities

An alarm will not be responded to blindly. They strongly recommend that any alarm have a corresponding interface that clarifies the alarm, its source, and enables an operator to confirm its validity. This recommendation is supported by Sorkin and Woods' model.

3. Make alarms unambiguous

The authors want to minimize representational errors with this recommendation and suggest that an alarm should somehow also direct user attention to causal factors.

4. Reduce false alarms

Much like Bransby and Jenkinson did, the authors also mark false alarms as a, if not the biggest, hindrance to operator SA. The authors likewise point out that one way to minimize false alarms is to give operators the power to modify the alarm limits:

"One approach to the reduction of false alarms is allowing people to tailor alarm limits, which may be appropriate in some domains. Medical equipment, for instance, often allows the physician to adjust the level at which a particular physiological monitor will alarm - she can set mean heart rate to alarm at 120 for one patient or 150 for another based on her knowledge of physiology (based on age, weight, physical condition and whether the patient smokes) for that patient. This situationally defined tailoring minimizes false alarms.”

5. Set missed alarm and false alarm trade-offs appropriately

The authors point out that, depending on the type of situation being alarmed, and the typical or anticipated workload of the operator, it may be better to allow some false alarms if only to prevent any missed alarms, or vice versa.

6. Use multiple modalities to alarm, but insure they are consistent

As Belz et al. and others have discussed, the combination of visual and audio cues make for more effective alarms than either alone. Endsley and her colleagues reiterate this finding with the recommendation that multiple modalities should be preferred when designing an alarm system.

7. Minimize alarm disruptions to ongoing activities

When designing an alarm system, never forget that the operators have many other tasks and responsibilities besides responding to an alarm. The authors put it succinctly: "In general, alarms need to inform people of the event in question, without requiring that they be distracted from current tasks unless they choose to be."

8. Support the assessment and diagnosis of multiple alarms

To help diagnose alarms, the authors identify three tasks an alarm support system should include: determining which alarms have triggered and which have not, which are new, and in what order they were triggered. They also advocate the addition of trend displays as well as grouping information related to or influencing each alarm.

9. Support the rapid development of global SA of systems in an alarm state

The authors state, "...it is important that operators are able to ascertain at a glance the status of the entire system to gain global SA." Providing an overview of the system, beyond just the alarm information is important for operator assessment activities.

Tools That Support Predicting Events

Alarms are a simple interface to support humans' functioning in a complex environment, but perhaps they are too simple. Endsley's model of SA treats awareness of current individual feature conditions as the lowest level of SA. The highest level is achieved when is it possible for the individual to predict impending situations and respond in their anticipation. This idea of forecasting has been explored in several fields, and interfaces and technologies to support it remain an active area of study. I discuss several areas of research that have explored the topics of forecasting as part of a monitoring task.

Industrial engineering has long studied methods of production process optimization. Two issues that have been of particular interest in those endeavors are quality control and equipment maintenance. These areas are actually closely related. In quality control, the goal is to detect when a portion of production is operating outside of tolerances in an unpredictable or unexpected way. Maintenance research seeks to optimize the time, effort, and cost associated with repair due to equipment failing, operating unpredictably, or producing outside of tolerances.

Statistical Process Control and Control Charts

In manufacturing, much of the process is predictable, but not entirely so. In those complex environments, where so many moving and ever-wearing components work with such a wide variety of source materials, there is significant variation from product to product. Shewhart (1931) classified the variation as belonging to two classes: natural or expected, which he called common cause, and uncontrolled, which he called special cause. As part of quality control monitoring, statistical process control (SPC) is used to monitor the variation of any aspect of a manufacturing process by using statistical tools to predict undesirable product variation. Of particular interest is the use of control charts and the application of SPC in tools to perform predictive maintenance

An important artifact used in SPC, and more importantly, in predictive maintenance, is the control chart. A control chart is used to identify when a process is operating outside of expected norms. It, as originally proposed by Shewhart (1931), consists of a set of readings plotted as connected points. The points are rendered with a line representing their mean reading, as well as two lines that represent their upper and lower control limits. Optionally, a control chart can include upper and lower warning limits and other zone delineations. Reading the chart involves identifying if any readings are outside the control limits, or if a pattern is apparent within the readings themselves. The purpose of a control chart is to facilitate detection and classification of events that reflect a change in a system's process or performance. There are several sets of rules for detecting a signal in a control chart (Nelson 1984), (Western Electric 1956). The rules, in each case, involve looking for features in the chart such as series of points near the control limit lines as well as patterns such as repetitions or trends.

Variations of the control chart exist to improve its sensitivity to small variation. The CUSUM (CUmulative SUM) chart sums each subsequent sample, which amplifies the effect of change in the samples. The Exponentially Weighted Moving Average chart, like the CUSUM chart, also takes more than just the most recent measurement into account to determine if a signal is present, but it reduces the weight of older readings in its calculation. (see (Montgomery 2008) for a more detailed discussion)

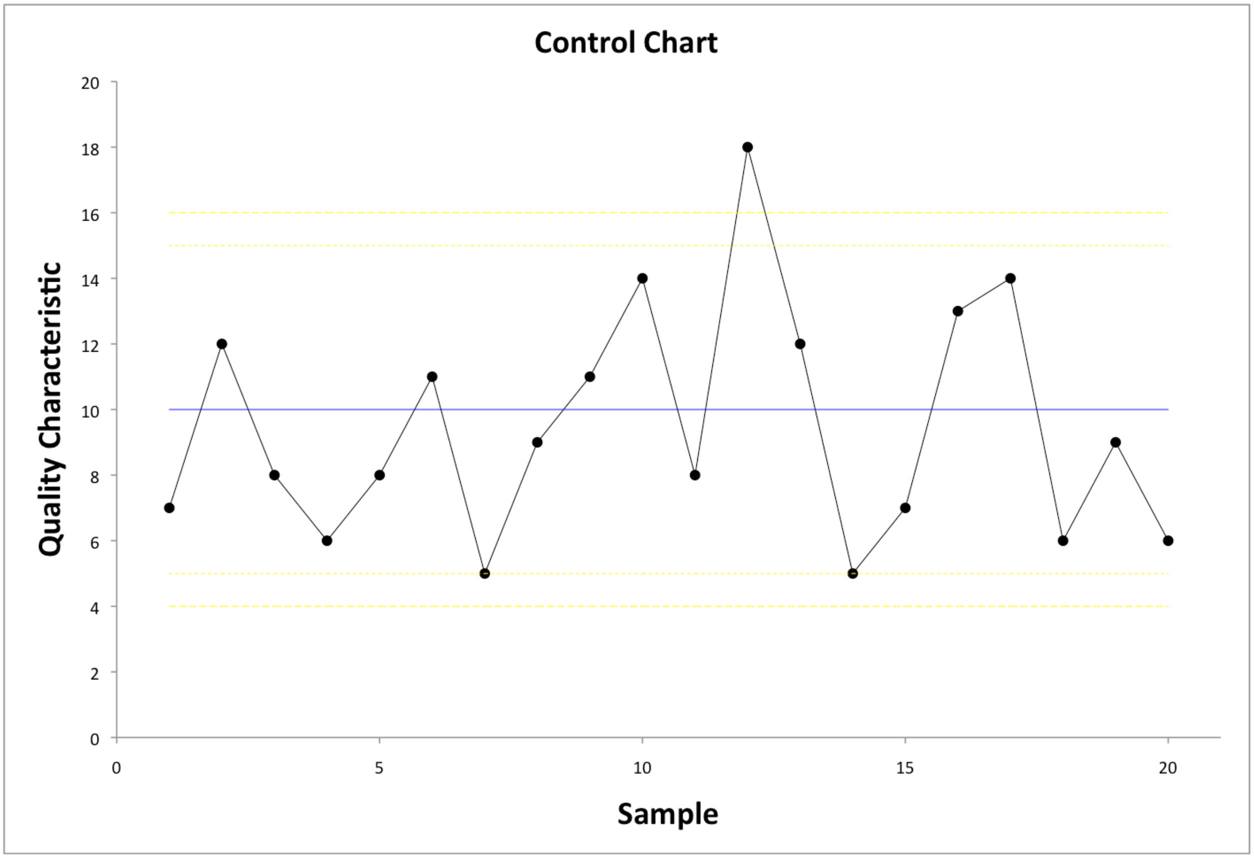


Figure 3: An example control chart

Predictive Maintenance

One of the goals of detecting process variation is to determine when to perform maintenance. However, determining the ideal time to perform maintenance on a piece of equipment is not easily accomplished. A simple heuristic approach commonly used is to schedule maintenance on a regular interval expected to be shorter than the mean time to failure for the piece equipment. This sort of preventative maintenance is common practice in many areas of industrial operations, and even in everyday life for many people (did you change your oil in the last 3,000 miles?). The problem with preventative maintenance is that it is costly; less costly than were the equipment to break, but wasteful because it brings production to a halt more frequently than absolutely necessary, as well as consuming the resources needed to perform the maintenance (time, labor, equipment) more than the ideal. In response, industrial engineering has devised techniques for performing predictive maintenance instead. Predictive maintenance arose out of the airline industry, where it was called Reliability-Centered Maintenance and was intended to optimize the maintenance requirements for aircraft (Nowlan and Heap 1978). Predictive maintenance (PdM), also known as condition-based maintenance, is based on a general methodology of sensing the condition of various equipment parameters, having a knowledge base of known failure conditions, forecasting the degradation of the equipment using statistical means of analysis to estimate when failure would occur, and only scheduling maintenance when that limit is approached. The focus of all PdM is first and foremost based on the ability to accurately and frequently sense the operating conditions of equipment. Some sensing techniques used in PdM include infrared imaging, vibration analysis, and oil analysis. Forecast analysis is then performed using a variety of established statistical methods, such as trend analysis and averaging. For an extensive review of predictive maintenance techniques and technologies see (Mobley 2002).

Recently, there have been two applications of predictive maintenance that are relevant to this dissertation. Work by Butler et al. has expanded the concepts of predictive maintenance to accommodate the most complex of maintenance applications, and work by Ni et al. has attempted to generalize PdM techniques by creating an intelligent agent system that can be used in a variety of environments.

Work-centered design (Butler, Zhang et al. 2007) acknowledges the reality of the maintenance of very complex equipment and tries to accommodate all the myriad of resources and requirements that limit the efficient execution of a maintenance task. The maintenance of aircraft doesn't just require the aircraft to be present at the maintenance facility, for example, but also the availability of a hangar, the appropriate skilled workers, and the necessary parts and maintenance equipment. In a complex operating environment like an airline maintenance hangar, making sure that all of those resources are available at the same time is a very challenging task. The notion of work-centered design is based on accounting for all of the necessary resources needed to complete a particular project, understanding the order of operations toward completion, and then synchronizing all of the requirements with already scheduled or active projects. Work-centered design is an example of an application that attempts to take into account all the aspects of a task in a bottom-up fashion.

Ni et al. (2003) developed a tool called Watchdog which applies multi-sensor assessment and prediction of equipment performance to a range of industrial equipment. The authors describe the functions of the Watchdog agent as being: "quantitative multi-sensor assessment of performance degradation, forecasting of performance degradation, [and] diagnosis of the reasons of the current or predicted performance degradation." Watchdog uses complex, domain-specific assessment and diagnosis methods to postulate when and what to perform maintenance work on. To accomplish this, Watchdog includes tools to perform signal processing and feature extraction, quantitative health assessment, condition diagnosis, and performance prediction. It then displays the results of these tools in dashboards consisting of many time-domain graphs of those performance metrics and their predicted values.

Tools For Learning About Complex Systems

A vital but often overlooked function that a human serves in a supervisory control mode is that of learning. In complex environments, conditions are constantly changing, and any person interacting with that environment is constantly learning and adapting to those conditions. As the person gains more experience, the manner by which they monitor the complex system changes. I discuss work that has explored how people learn in and about complex systems, as well as previous work that has attempted to capture what is learned.

Learning In and About Complex Systems

Sterman discussed the need for tools to support learning in and about complex systems (Sterman 1994). Sterman bases his discussion on a feedback learning model whose origin he traces back to Dewey, in which the real world provides information feedback to the individual, who in turn uses the information to make decisions which then influence the real world. He stresses the role of mental models, how they are informed by the information feedback, and how they in turn influence the strategies used to make decisions. He concludes that for individuals to successfully learn in and about complex systems, tools that elicit and record those individuals' knowledge are necessary, as well as tools that help them record what they've observed and how they expect the system to respond. Sterman identifies many factors that inhibit learning about the system while interacting with it. In particular, he recognizes factors associated with perception and factors associated with mental models and reasoning as the primary sources of interference. Dynamic complexity, limited information, confounding variables and ambiguity, and misperceptions of feedback impede the perception part of the feedback learning model. Flawed cognitive maps of causal relations, erroneous inferences about dynamics, unscientific reasoning, defensive routines, and implementation failure interfere with the feedback and reasoning aspect of the feedback learning model.

Sterman concludes that a tremendously valuable tool for learning about complexity is the use of simulations that allow the decision maker the opportunity to "…refresh decision making skills, conduct experiments, and play" (Sterman 1994). In fact, he suggests that only simulation will do, because "without simulation, even the best maps can only be tested and improved by relying on the learning feedback through the real world …this feedback is very slow and often rendered ineffective by dynamic complexity, time delays, inadequate and ambiguous feedback, poor reasoning skills, defensive reactions, and the costs of experimentation. In these circumstances simulation becomes the only reliable way to test the hypotheses emerging from elicitation techniques and other problem structuring methods." This conclusion is troubling because simulation is often not only difficult, but nigh impossible for many complex systems.

Fuzzy Learning Applications

Ren and Sheridan (1994) combined the imprecise nature of expert knowledge with a more precise operating model using methods from fuzzy logic. They used fuzzy linear programming to model the rules used to guide an automated task, namely freight train dispatching, using a fuzzy knowledge base that was built using the imprecise language and expert knowledge. They demonstrated that building such a knowledge base was possible and could lead to an improvement in an automated task. They next built on this work (Ren and Sheridan 1995) and added support for a human in the loop who is able to adapt the system to changing conditions, thus showing that a user can interact with a fuzzy knowledge base to improve a decision-making process. However, the manner by which they accomplish this is through the manipulation of a goal satisfaction criterion, which, as an interface, they do not describe.

Buharali and Sheridan (1982) focused on the dynamic construction of the fuzzy knowledge base by tracking how a subject modified rules and parameters in the dynamic control task of following a path. Their technique acquired the rules and parameters on the fly as the human realized the need to add new information. The system encouraged the human supervisor to elicit new rules that it identified as missing using an inference engine.

These applications of fuzzy logic demonstrate that users can contribute to a knowledge base and that it is possible to elicit expert knowledge even in situ using fuzzy logic as the basis.

Designing Monitoring Tools

Several authors have discussed, in broad terms, how monitoring interfaces create poor user experiences and have offered design guidelines for improvement. Their writings have provided guidance in the design of the work discussed in this dissertation.

Communicating with Automation

In 1990, Norman articulated one of the most significant problems facing an increasingly automated operating environment in so many industries: "…automation is at an intermediate level of intelligence, powerful enough to take over control that used to be done by people, but not powerful enough to handle all abnormalities. Moreover, its level of intelligence is insufficient to provide the continual, appropriate feedback that occurs naturally among human operators." (Norman 1990). This assessment could be applied even today, almost twenty years later. In fact, in 2007 Norman expanded on this concern by offering the following advice: "As we move toward an increasing number of intelligent, autonomous devices in our environment, we also need to transition toward a more supportive form of two-way interaction. People need information that facilitates discovery of the situation and that guides them in deciding how to respond or, for that matter, reassures them that no action is required. The interaction has to be continuous, yet nonintrusive, demanding little or no attention in most cases, requiring attention only when it is truly appropriate. Much of the time, especially when everything is working as planned, people only need to be kept in the loop, continually aware of the current state and of any possible problems ahead." (Norman 2007) I treat these as design guidelines for creating monitoring interfaces. The goal of the work described in this dissertation is to provide a framework to convey which situations may require attention and which do not, and to do so continuously, without disruption, by developing situation awareness. The approach will be to expose the alarm processing continuously to the system observer.

Chapter 3: Heed and Its Arithmetic

Heed Defined

I define "heed" as the importance of attending to a situation. I assign 0 to the minimum importance of attending to the situation and 1 to the maximum. Heed is situation-dependent and not data source-dependent, and a numerically higher data value does not necessarily mean a higher heed value.

Heed can be described in terms of statistical process control. In quality control assurance, an out of control process requires a quality control engineer's attention. A process is considered under control when the variation of a quality control parameter hovers about an expected mean value, and while it is within the control limits defined for that manufacturing process. When that quality control parameter exceeds the predefined control limits, then that process is classified as out of control, and a quality control engineer is expected to take action to repair the equipment (Deming 1975). Heed can also be thought of as the range from the expected mean value of a process to either one of its control limits. The closer a process is to being out of control, the more important it is to attend to it; the closer it is to an expected value, the less important it is.

The first step in transforming any sensor reading from its native scale into the heed scale begins with defining the two sensor reading thresholds that identify the boundaries of "safe to ignore" and "must attend." I refer to these as the ignore-boundary and the attend-boundary respectively. The ignore-boundary sets the sensor threshold for "minimum" heed, and the attend-boundary for "maximum" heed. The range between the boundaries describes a "sensor condition" such as a gas tank being empty, the speed of a car being slow, or a refrigerator being warm, and is defined by, minimally, a reporting sensor and the two boundaries. Attend and ignore are useful terms for thinking of the boundaries in terms of user activity, but maximum and minimum are useful terms for thinking about what the application author wants the display to show. Likewise, unexpected and expected, or abnormal and normal are useful for thinking in terms of an expert's knowledge of the system.

The second step in calculating heed from sensor data is defining a transformation function, H(t), for the readings between the boundaries. Frequently, a linear transformation is sufficient.

In situations where heed increases when the sensor value gets too large, I define the transformation from sensor data to heed as follows:

$$heed(t) = \begin{cases} 0.0, & if\ sensor(t) < H_{\min} \\ H(t), & if\ H_{\max} < sensor(t) < H_{\min} \\ 1.0, & if\ sensor(t) > H_{\max} \end{cases}$$

In situations where heed increases when the sensor value gets too small, I define the transformation from sensor data to heed as follows:

$$if\ H_{\min} < H_{\max}$$

$$heed(t) = \begin{cases} 0.0, & if\ sensor(t) < H_{\min} \\ H(t), & if\ H_{\max} < sensor(t) < H_{\min} \\ 1.0, & if\ sensor(t) > H_{\max} \end{cases}$$

$$if\ H_{\min} > H_{\max}$$

$$heed(t) = \begin{cases} 0.0, & if\ sensor(t) < H_{\max} \\ H(t), & if\ H_{\min} < sensor(t) < H_{\max} \\ 1.0, & if\ sensor(t) > H_{\min} \end{cases}$$

$$H_{\min} \neq H_{\max}$$

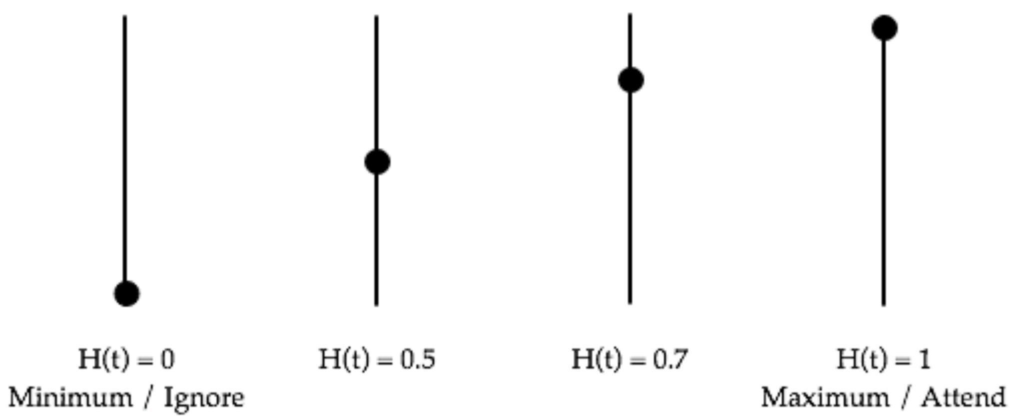


Figure 4: Four heed values and matching visual representations.

Sensors need not be continuous. A lookup table that maps discrete values to heed could be used for H(t) to transform non-numerical data, such as the appearance of certain terms in a stream of text, to values between 0 and 1. (Some abstract sensor examples: a sensor that reports the appearance of "danger" words used in a public forum and maps them to heed: "bomb": 0.9, "poison":0.6, "machine gun": 0.7. Such a sensor would allow an observer to only pay attention to the stream when those terms were being used.) As long as the data is reported with one dimension in time, and as long as a function H(t) can be defined to transform the readings into heed, then data recorded can be used as a sensor.

Heed Notation

I use basic set notation to discuss heed and, unless an additional transformation function is specified, a linear transformation is assumed. The notation is as follows: {m:Hmin;M:Hmax}units or {m:Hmin;M:Hmax | H(t)}units.

Sensor Conditions As Fuzzy Sets

The definition of a sensor condition comes from fuzzy set theory. Calculating heed is based on the methods of calculating membership in fuzzy set theory. Membership functions are defined by Zadeh as:

"A fuzzy set (class) A in X is characterized by a membership (characteristic) function fA(x) which associates with each point in X a real number in the interval [0,1], with the value of fA(x) at x representing the grade of membership of x in A. Thus the nearer the value of fA(x) to unity, the higher the grade of membership of x in A." (Zadeh 1965)

Where fuzzy set membership functions encode a truth value, mu, sensor conditions encode an importance value, namely heed. To be more accurate, fuzzy set membership encodes the degree to which a characteristic is met. If a fuzzy set is defined for "tall", then membership encodes how true the statement "a person of height x is tall" is. Similarly, heed can be defined in terms of fuzzy set membership, and encodes how true the statement "a sensor reading of x indicates a situation is important to attend to" is. Using this definition, heed can be seen as an application of fuzzy set theory to the activities of user interaction.

I transform fuzzy set theory into heed arithmetic with some practical guidelines. In fuzzy logic, an application of fuzzy set theory, set membership is used as an antecedent in rules and the degree of membership in each set is mapped to a truth value to determine the truth of a rule statement. The rule is de-fuzzified, also known as making the value crisp, by turning the fractional truth value into a decisive true or false, which in turn is applied to an automated control system. Heed, on the other hand, is never made crisp. The fractional heed value is what is displayed or communicated to a system observer. Heed is specifically intended to be a value communicated to human observers, operators, and end users to facilitate making decisions - it is not intended to be applied as a means for automation.

Another practical difference between heed arithmetic and fuzzy set theory is that fuzzy sets are frequently represented by three or four boundary conditions, but sensor conditions are typically only represented by two. For example: a fuzzy set for “warm” might be false if temperature is below 50°F and above 90°F and true between 60°F and 80°F. However, a sensor condition would typically only be defined as 0 below 50°F and 1 above 60°F. This distinction may seem unusual, but it stems from the practice of setting alarms. Typically, an alarm is set to trigger beyond a threshold, and thus that threshold divides a sensor range into two parts: the range when the alarm is triggered, known as onset, and the range when it is not, known as offset. A sensor condition is a membership function that is defined over the offset range adjacent to the onset range, and so the only boundaries that need to be defined are the two that describe where the onset begins and the offset ends.

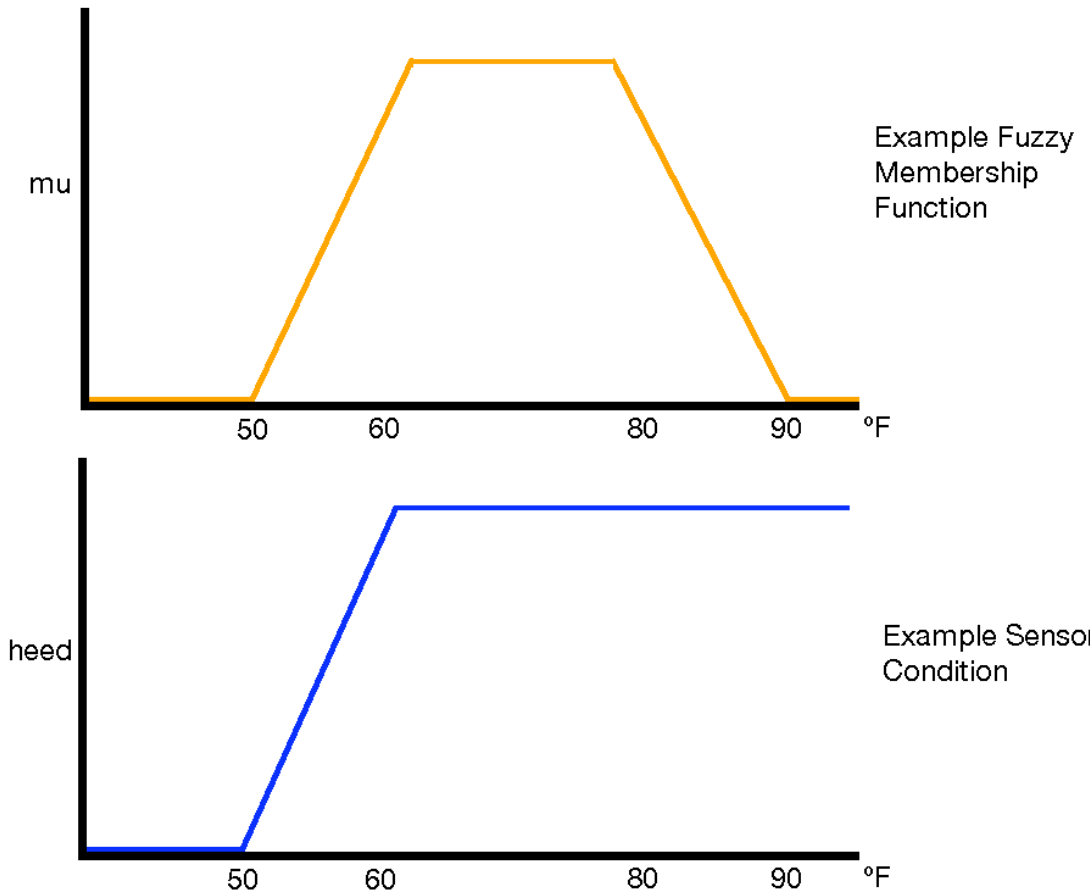


Figure 5: A typical fuzzy membership function and a typical sensor condition

Although in practice heed usually increases or decreases with sensor value, this is not always the case. For example, if a tire pressure monitoring system were designed with a heed-based display, then a driver may want to define "normal" pressure as having a heed of 0 between 30 and 34 psi, and a heed of 1 for pressure below 26 psi or above 38 psi. However, the same result can be achieved by defining two sensor conditions: one for pressure too low and one for pressure too high and then combining them into a schema, as described below.

The Heed Value For A Schema

The previous chapter discussed situation awareness and the role monitoring plays in establishing and maintaining SA. Endsley describes mental models as being a powerful aid in the establishment of SA by helping to quickly classify the situation (Endsley 1995). She discusses the importance of schemata as short cuts to recognizing situations without having to exercise a full mental model by enabling the individual to recognize only the prototypical states of the situation. To support this activity of situation classification, the heed of sensor conditions can also be combined into a schema. Again, the techniques established in fuzzy set theory are applicable here; in this case, the techniques of fuzzy set combination. What follows is a discussion of how to compute logical combinations of heed values by adapting methods from fuzzy set theory. The most significant difference between heed and fuzzy logic, other than the lack of a de-fuzzification-like process, is how the logical operators are applied. Heed, at this time, only supports a small subset of possible logical operators, and not all of these operators behave like their fuzzy logic counterparts. The operators that are included in heed arithmetic are chosen based on how their application affects the resulting heed and how a heed display author writes heuristic rules. The use of logical operator terminology is intended to support schema or rule authoring, but should not be mistaken for a full-fledged logical framework. Providing a completeness proof or adding additional operators not discussed in this dissertation, such as NAND, NOR, modulo, or remainder is left for future work. Each logical operator in heed arithmetic presented here is prefaced by a discussion of the monitoring situation for which that operator is applicable. This format is intended to provide a framework to heed display designers for how and when to apply which operator, and to justify why not all known logical operators are included in this discussion; the operators discussed are not provided to create a full logical framework, but instead, to support observer activities or designer goals.

Monitoring For Abnormal Activity And The NOT operator

A common monitoring goal is to identify a situation wherein a system is operating abnormally. Such a situation necessitates some knowledge of normal activity first. Consider this example of an automobile's engine temperature: normally it runs at 190°C, and almost never over 210°C. If we know the normal operating range, then the appropriate time to attend to the situation is when the engine temperature is NOT normal. Thus, heed arithmetic requires an operator akin to the complement operator; i.e. the NOT operator.

In fuzzy set theory, calculating the membership of a complement, such as NOT $\mu(t)$, is accomplished by taking 1 minus the membership of $\mu(t)$. Calculating the complement of a heed value $h(t)$ is accomplished likewise, by taking 1 minus $h(t)$. In practice, when defining a sensor condition, the system observer identifies when a sensor reading warrants attention and when it does not. By swapping the two boundaries, and reversing the transformation function, the same result as the complement operator is achieved.

When defining the normal engine temperature sensor condition above, the nominal value of 190° would have to be assigned to Hmax, i.e. the attend boundary, and 210° to Hmin, the ignore boundary, which is counter-intuitive. It seems strange to describe a situation wherein we must attend to something when it is usual, but ignore it when it is unusual. It would be more natural to define the sensor condition by reversing the boundary values. This example also illustrates why it is sometimes more valuable to think of the sensor condition boundaries in other terms, such as normal and abnormal instead of ignore and attend. Despite being able to achieve the same result as the complement operator by reversing the boundary values, the complement operator is still useful because it can be applied to the result of more complex heed schema. A more complex heed schema can be used to define what is normal or expected in terms of many sensor conditions, and the complement operator can then express the importance of attending to an unusual situation.

Figure 6: A comparison of the truth table for Boolean NOT and heed NOT operators

Monitoring For Dependencies And The AND Operator

Some situations are defined by a chain of dependent conditions. The situation does not arise until all the dependencies are met. For example, consider monitoring the driving performance of a car based on its tire pressure and its speed, where lower tire pressure limits maximum speed. As tire pressure goes down, the speed at which the vehicle loses control goes down as well. The performance of the car is based on these two interdependent factors. In order to describe situations such as this in terms of their dependencies, a depends-on combination operator is needed; i.e. an AND operator.

In fuzzy set theory, the AND combination of two sets (their intersection) is the minimum of the two. Using the minimum of two truth values ensures that the fuzzy-AND and the arithmetic-AND behave the same: as long as either A or B are not true, the combination is not true. In heed arithmetic, the goal is to identify when to attend to a situation and, where possible, to do so before the situation has arisen. The minimum of two heed values would mean that, until both dependencies are met, the situation would be ignored. Instead, the preferred way to think about this in terms of heed is that once one of two dependencies is met, then the situation is halfway to fruition and is already somewhat important to attend to. Therefore, a compensatory measure of AND is a more appropriate calculation than a minimum measure. To illustrate why a compensatory measure of AND is preferable, consider again the example of monitoring the driving performance of a car based on its tire pressure and its speed. As tire pressure goes down, the speed at which the vehicle loses control goes down as well. Imagine a scenario wherein we are driving on a winding mountain road behind a slow-moving truck. Our tires are a little low, but we have excellent control at the low speed we're moving. Our car has a heed display; one that tells us to take heed of a potential out-of-control situation wherein our tire pressure is low and our speed is high. First, let's consider what the display would show were we to employ the fuzzy set AND operator which shows us the minimum of the two heed values. Our tire pressure is low enough that the sensor condition for 'low tire pressure' returns a high heed value of 0.8, but the sensor condition for speed returns 0 since our speed is too low for it to be of any concern. Our display, were it to show us the minimum of the two values, would indicate a heed value of 0 in that case.

As we round a corner, we can see the next 300 yards of open roadway and decide to pass the truck. We jam on the accelerator, and as we begin to pass the truck, suddenly the heed display jumps to a high value! This occurs because, using the minimum of the two conditions, the speed condition dominated the result of the calculation prior to passing. As we began to pass and approach the maximum of the 'high speed' sensor condition, it rose quickly, and since the speed sensor dominates the calculation, the heed value of the combination rose just as rapidly. Instead, we could have used some indication that, even though we weren't in any danger prior to passing, the situation was ripe for danger. The fact that the tire pressure was already low should have given us some cause for concern, even at low speed. A compensatory measure such as the mean of the two sensors conditions would have been more appropriate.

A compensatory measure would take into account any contribution from either sensor and, like the arithmetic AND calculation, result in a 0 value if both are 0 and a 1 if both are 1, but would be between 0 and 1 if either value were between 0 and 1. In this dissertation, the mean of two heed values is used to calculate the AND of the two values. Compensatory measures exist in the fuzzy logic literature as well; for a detailed discussion see (Zimmermann and Zysno 1980).

Had the vehicle's heed display used a compensatory measure of AND in the previous example, prior to passing the truck when the speed was low and the reduced tire pressure is not a significant factor in the vehicle's handling, the heed value of tire pressure AND speed would be somewhere around 0.5 and would thus indicate the potential that a situation (in this case, loss of handling control) might need our attention soon. In a situation defined by dependencies, i.e. one with an AND, the more dependencies are met, the more likely it is that the situation may arise soon.

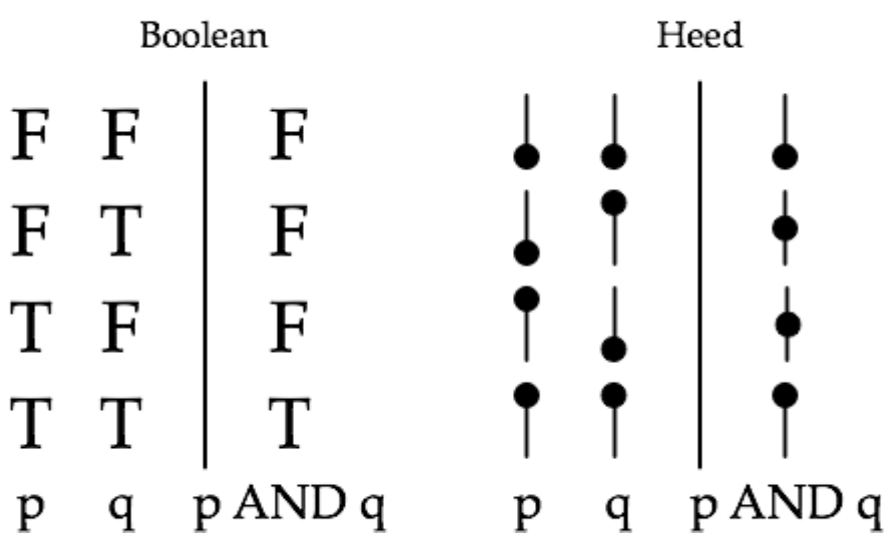


Figure 7: A comparison of the truth table for Boolean AND and heed AND operators

Filtering Monitoring Tasks And The OR Operator

Complex systems require observers to monitor for many situations. The more situations that must be monitored, the more it falls to the display designer to figure out how to display all those situations without it becoming cluttered or otherwise overwhelming the user. One way to reduce display clutter is to provide an operation that combines situations. Once combined, the resulting display should indicate if any of the constituent situations requires attention. The OR operator enables the combinations of situations.

In fuzzy set logic, the union, also known as the OR, of two sets is the maximum membership of either set. In heed calculus, it is the same, with the maximum of constituent heed values representing the combination. The maximum is chosen because the OR combination of two heed values is intended to convey to an observer that at least one of the constituent situations should be attended to, as well as the importance of attending to that situation, and to convey that the others can either be ignored or that it is not as important to attend to them. Furthermore, this combination operator is best applied to situations related to one another; i.e., situations that constitute a category of situations to be monitored for. For example, by combining a set of situations related to networking and then creating a separate combination of situations related to disk usage, the resulting display would enable the observer to quickly decide if the disk or the network needs attention.

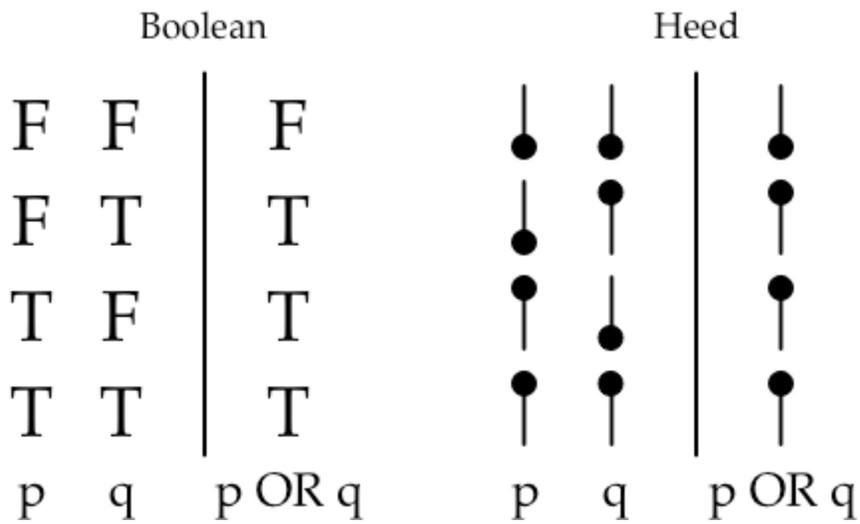


Figure 8: A comparison of the truth table for Boolean OR and heed OR operators

In addition, the OR operator can be used to obtain the same result as defining a sensor condition with four points instead of two. Consider again the example of a tire pressure monitoring system, where normal pressure is between 28 and 34 psi, but abnormal pressure is either above or below that. The sensor condition could be defined by four points, with a heed of 1 below 26 and above 38 psi, and with a heed of 0 between 28 and 34 psi. Alternatively, the same situation could be captured by defining two sensor conditions: one for pressure too high {m:38;M:34}psi, and one for pressure too low {m:26;M:28}psi, and then combining them with an OR operator, resulting in the English language equivalent of "tire pressure is abnormal when pressure is too high or pressure is too low."

Mitigating Monitoring Tasks And The AND-N Operator

One of the problems of an alarm is that it can be triggered despite the existence of a mitigating condition. For example, consider an alarm that is triggered when remaining disk capacity on a server drops below 5%, but does not take into account that during nightly backup the disk fills a temporary folder with cache files for the backup tape drive to spool out, a network disconnect alarm that is triggered during nightly server reboots or during scheduled downtime, or even a bedroom alarm clock that sounds on weekends and holidays. The ability to add mitigating circumstances to the description of a situation is vital. For heed, mitigating conditions should act as heed dampers, lowering, or even fully suppressing heed when mitigating conditions are met.

There are two ways to think about the addition of mitigating circumstances within heed arithmetic. The first is to think of a heed combination as an if…then rule statement, and mitigating circumstances add an …unless clause. Such a control structure is used in several programming languages, most notably in Perl (Wall 2000). A heed calculation can be thought of as following a control structure akin to the English equivalent of "if (combination of monitored heed conditions is high) then attend that situation." By adding an unless clause the rule becomes "if (combination of monitored heed conditions is high) then attend that situation, unless (combination of mitigating heed conditions is high)."

The second way to think about the addition of mitigating circumstances is with an operator I call AND-N, which is short for AND-NOT. The English equivalent for AND-N is "if (combination of monitored heed conditions is high) AND NOT (combination of mitigating heed conditions is high) then attend that situation." This too allows for the mitigation of heed conditions by negating and joining via an AND operator. However, the AND-N operator is neither like the Boolean NAND operator, nor is it the same as taking the complement and then applying the heed AND operator. The NAND Boolean operator returns true unless both operands are false. That would be the equivalent of heed being high when both conditions are low; instead the result we want is one that is high when the monitored heed condition is high and the mitigating heed condition is low, and is low otherwise. Furthermore, applying the previously discussed heed AND operator, which acts by averaging the heed of the first set with the mitigating set, would not give the desired result of fully suppressing heed in the presence of the mitigating condition. Instead, I propose an operator that returns the minimum of the first condition and the complement of the mitigating condition.

I've already discussed the minimum of two heed conditions during my discussion of the AND operator and why, in heed arithmetic, it is not the appropriate calculation because it suppresses high heed conditions until all dependent conditions are high heed as well. However, when mitigating a high heed, suppression is exactly the outcome we want. By first negating the heed of the mitigating circumstance, the result of the AND-N operation, is a heed value that reflects the primary condition unless the mitigating condition has a high heed.

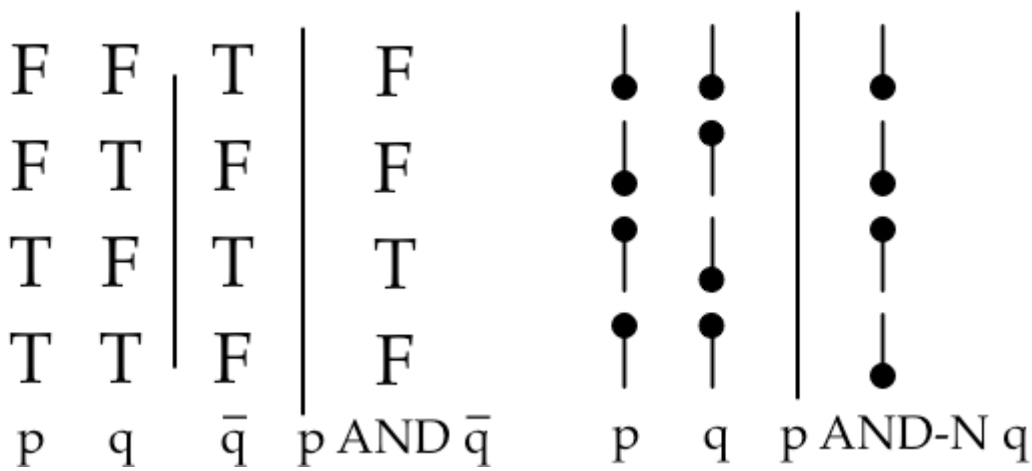


Figure 9: A comparison of the truth table for the Boolean operation p AND (NOT q) and the heed AND-N operator.

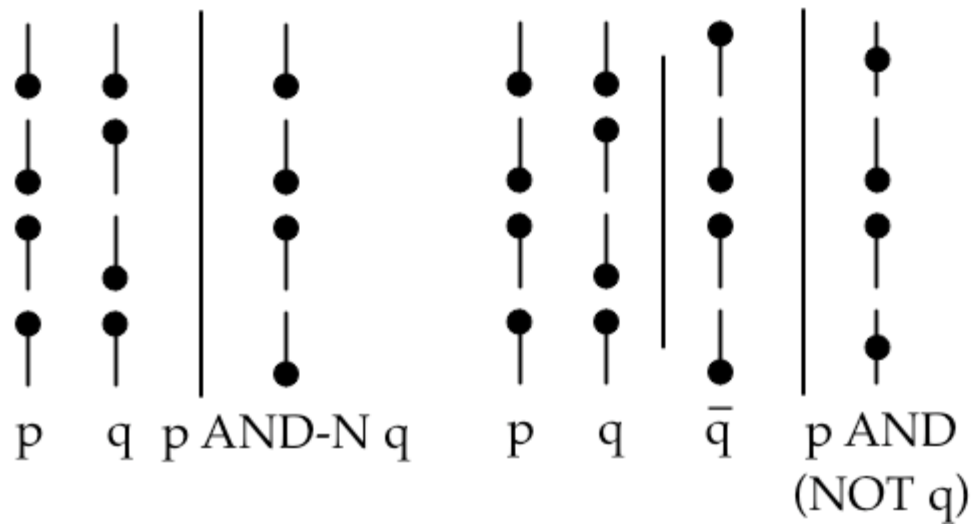


Figure 10: A comparison of the heed table for the heed operation p AND (NOT q) and the heed AND-N operator.

Transformation Functions

In the first chapter I briefly mentioned that transformation functions also play a role. For example, consider a fuel gauge: if the sensor condition is at minimum heed when the tank has 3 gallons remaining and at maximum heed when the tank has 0 gallons remaining, how should the scale convey heed in between? Should a half gallon change between 2 and 1.5 gallons produce the same change in heed as a change between 1 and 0.5? Instead of a linear transformation function, for this example, an exponential transformation function would be better suited.

Membership functions in fuzzy set theory are modified using what are referred to as hedges. Hedges are a way of strengthening or weakening a fuzzy set definition statement using adverbs. For example, a person who is 6' in height may be "tall", but not "very" "tall", while a person who is 6'5" is both. Hedges modify membership functions by replacing the function between membership boundaries. For an extensive discussion of hedges see (Zimmermann 2001). Given the relationship between fuzzy sets and sensor conditions, naturally, sensor conditions can be modified similarly by replacing H(t) with another function. For heed, this is useful because it enables the application author to convey different degrees of heed throughout the range of sensor readings. By using a logistic function for H(t), for example, changes near either boundary can produce a greater change in heed than in the middle, or by using an exponential function changes at one boundary can produce a greater change in heed than at the other.

Sampling

Heed can support arbitrary precision, but what is the value of precision when making a decision about what is important to attend? Is it useful to know that the condition of your snowblower has a heed of .7328 and that the condition of your fireplace flue has a heed of .7651? Conveying precision to a user in this case would be misleading. The application of heed is based on estimation of under what conditions an observer believes they should attend to a situation. It is not precise analysis. One way of mitigating misinterpretations of this sort, due to precision, is to downsample the heed values into a lower resolution.

Careful consideration also should be given to the frequency of updates to the heed display; i.e. the sampling of data in the time domain. How frequently should a display designed for our snowblower's condition update? Every second? Every hour? Every day? Even though the heed of its condition is changing a little bit every second, it doesn't change that much over the course of a day. Therefore, if we were to build a display to convey the heed of our snowblower's condition, we might be best served by updating it only daily. In this case, we decide how frequently to sample heed in time based on an estimate of a period over which we might expect the display to show a perceivable change. What constitutes a perceivable change is itself determined by our prior selection of range sampling. Digital signal processing has been dealing with issues related to sampling and subsequent signal detection for many years. Some discussion of these issues, such as selecting sampling windows, is in (Feuer and Goodwin 1996)

Forecasting Heed

Heed-based interfaces are intended to support situation awareness. In order to support the highest level of SA, level 3 SA; when an individual can predict what the situation will be, heed-based interfaces should incorporate forecasting. Forecasting can stem from the recent history or from historical pattern matching, but should be performed on the sensor data prior to transformation into heed values. Forecasting transformed values can obscure early warning as shown in Figure 11.

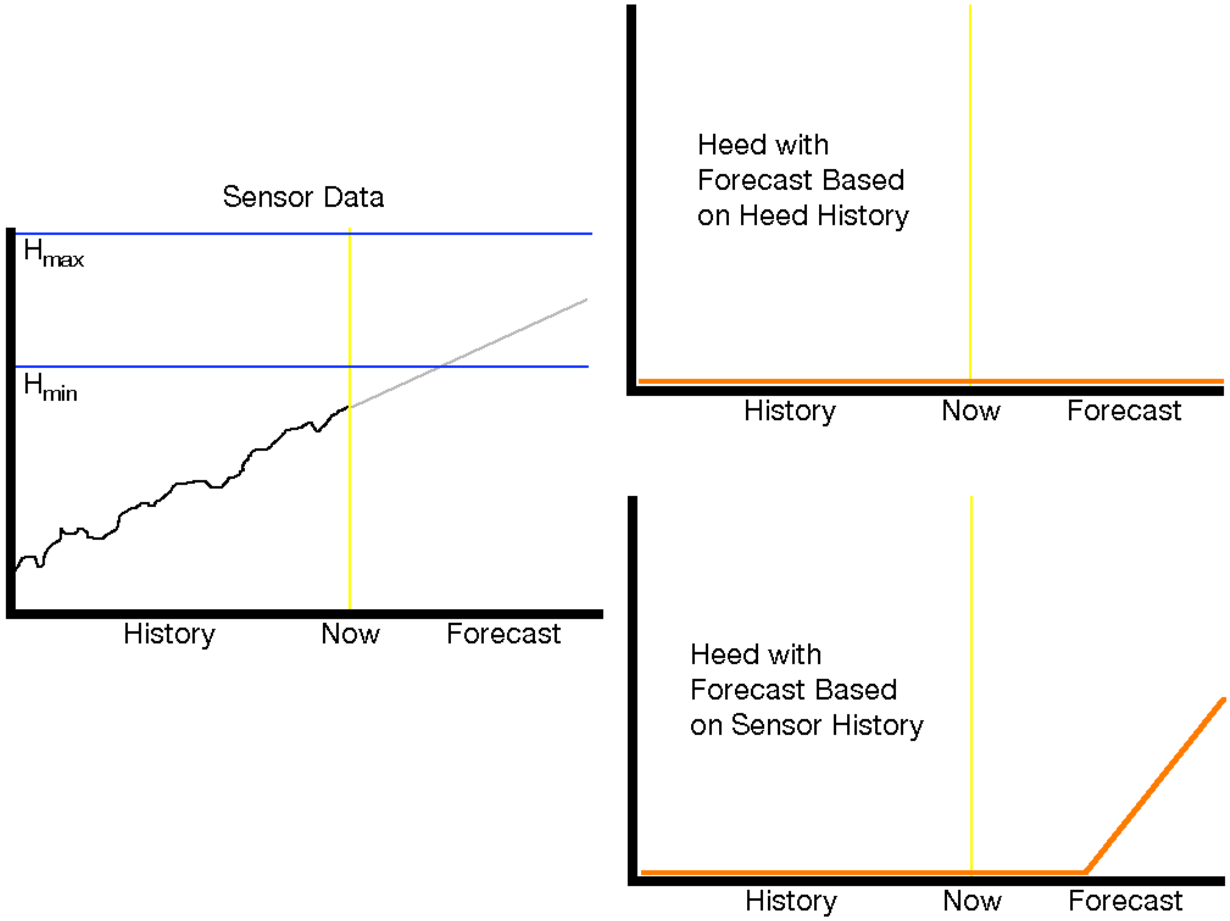


Figure 11: A comparison of forecasting heed based on recent heed as opposed to recent sensor data.

There are many different methods of calculating a time-series forecast. Numerical methods based on recent history include extrapolation (Brezinski and Zaglia 1991), regression analysis (Draper and Smith 1998), and Kalman filter (Simon 2006). A comparison of modern trend estimation from the financial forecasting literature can be found in (Bianchi, Boyle et al. 1999). Methods for forecasting based on historical pattern matching include reference class forecasting (Kahneman and Tversky 1979), and pattern informatics (Rundle, Tiampo et al. 2002). Exploring which methods of forecast yield level 3 SA and comparing their relative effectiveness is left for future work. For the work in this dissertation, forecasting is accomplished with a Kalman filter.

A Kalman filter is statistical method of estimating the state of a dynamic system. The filter estimates future states using a model built from a Markov chain of prior state changes discretized over time. A significant advantage of a Kalman filter is that it results in a distribution of likelihoods instead of a mean value and an associated error range. This probability distribution function lends itself well to visualization as will be demonstrated in the next chapter.

Chapter 4: Gizmometer: A Heed-Based Interface

Introduction

To explore how users respond to an interface designed to communicate heed as well as study how to author the underlying schema, an example interface was developed. The interface, which I call a gizmometer, is made up of a descriptor file and an interactive display. Together, they support three activities related to the heed of a situation being monitored for: authoring, perceiving, and delving. The authoring of the situation is supported in two forms: describing the situation and modifying the relevant sensor conditions. Perceiving the heed of a situation is supported by a novel display that presents heed in two views of differing detail. Finally, delving activities that support clarifying the heed of the situation are supported through a variety of display interactions.

I distinguish two primary roles of the individual involved in the monitoring task: making decisions of how and when to monitor a situation (as an observer) and interacting with the interfaces that facilitate that goal (as a user). Throughout this chapter I use term ‘user’ when discussing what action a person takes when interacting with the interface, and ‘observer’ when discussing the underlying tactics and motivations for those actions.

Authoring: Situation Description

An authoring interface is required to allow an observer to specify heed conditions. A simple though somewhat inelegant form of authoring interface is a text editor when it is used to modify a configuration file. In many cases, a domain-specific tool is much better suited to the task of authoring a complex source file, but a text editor and a structured configuration file can be sufficient. For purposes of this work, I have built the prototypes of the gizmometer interface using a text file containing a hierarchy as the underlying situation descriptor, and I leave the development of more advanced authoring interfaces for future work.

The situation description file is a hierarchy composed of six element-types recorded in Extensible Markup Language (XML). A schema is one way to describe a situation using simple set arithmetic, and a binary tree is one way to represent a schema by setting operators to nodes and conditions to leaves. The hierarchy must also include provisions to identify where sensor information is recorded and to parameterize each relevant sensor and sensor condition as well as the display itself. Thus, the six elements of the hierarchy used to describe a situation are: sensor databases, the sensors themselves, transformation functions, sensor conditions, schemas, and which schemas to display and how.

Sensor Condition

A sensor condition describes the simplest possible situation that an observer may wish to monitor, and consists of a reference to the sensor definition that identifies a particular sensor, the sensor readings that define the attend boundary and the ignore boundary, and the transformation function. The minimal sensor condition consists of a transformation function and a pair of thresholds. Visual features, such as sampling resolution should the condition be rendered separately, can also be recorded with the sensor condition.

Sensor Schema

A situation is rarely identified by only one sensor condition, but rather is usually described by a set of conditions in a logical combination. This is equivalent to fuzzy set combination and implemented here as a schema of sensor conditions. The smallest schema consists of a single sensor condition and, beyond that, any combination with other schemas using AND, OR, AND-N, and NOT. The schema is recorded as a series of pairwise combinations represented as a binary tree with sensor conditions as leaves and operators as nodes.

Sensor Database

Sensor data collected from any source is assumed to be stored in a database, and in some circumstances, several databases. Therefore, one element of the hierarchy is a description of each relevant database and how the application can identify and connect to it.

Sensor Definition

Many different sensors may be involved in the monitoring of a situation. Each sensor definition consists of a sensor identifier or name, as well as the bounds of possible sensor readings. Preferably, sensor definitions are provided by the creators of the system being monitored.

Condition Transformation

The transformation function, which was previously described as being similar to the hedge function in fuzzy set theory, must be identified in the hierarchy so that the sensor condition can reference it. The description in the hierarchy maps an identifier that the sensor condition can reference to the name of the function used by the data processing component of the application.

Display

Finally, which schemata are to be displayed by default are listed along with their names or identifiers that observers will find meaningful. In addition, a Universal Resource Locater field is provided to link each display with external monitoring and analysis tools that an observer can launch from the interface to investigate the situation fully.

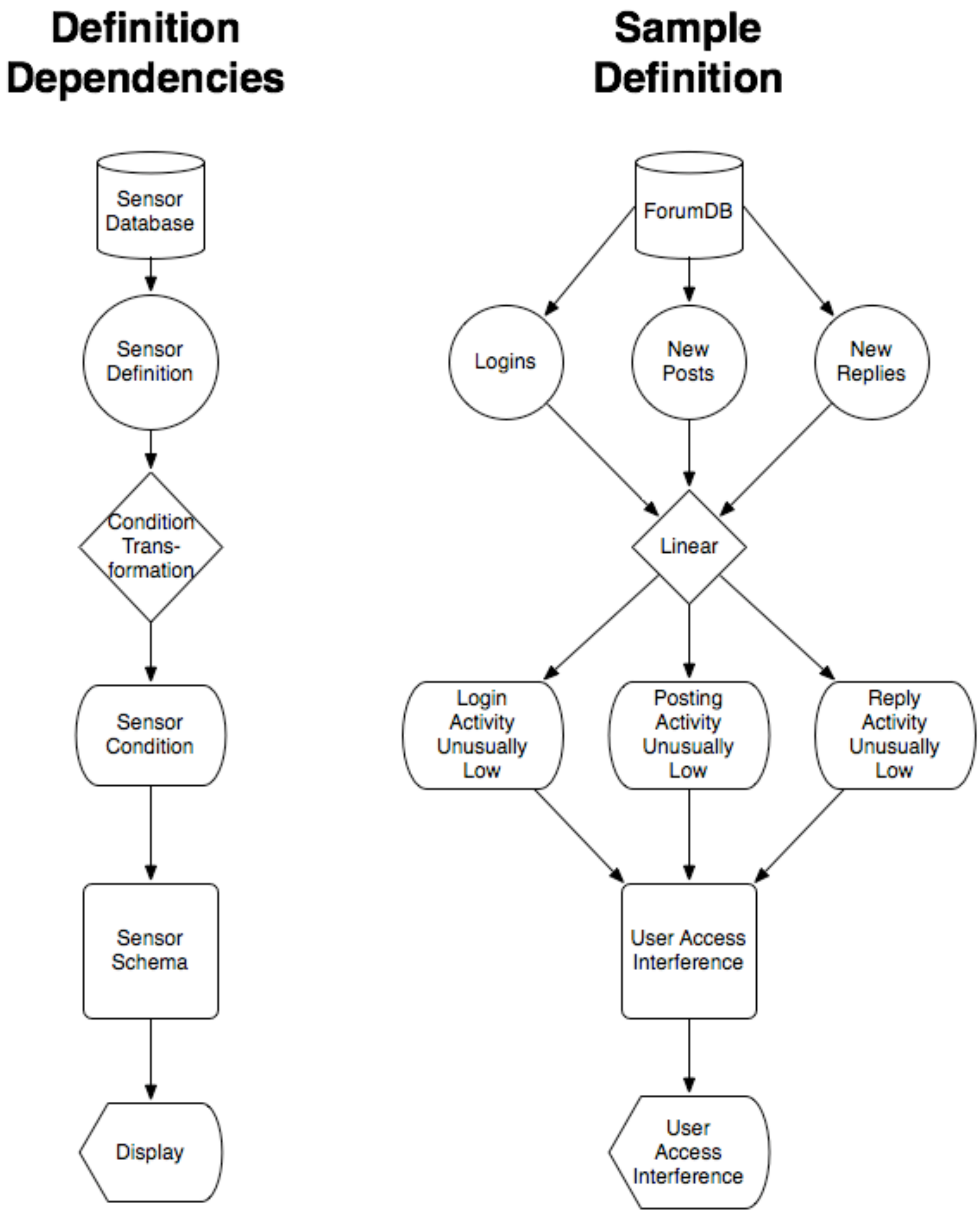


Figure 12: Dependencies in the situation description hierarchy and those dependencies illustrated for the example in Figure 13.

(Login Activity Unusually Low OR (Posting Activity Unusually Low AND Reply Activity Unusually Low))

```
<?xml version="1.0" ?>
<gizmometer-config>
    <sensor-databases>
        <database description="ForumDB"
            database="website_forum_db"
            location="sample.gizmometer.com" type="MySQL"
            user="heed_user" password="the_db_pass"/>
    </sensor-databases>
    <sensor-definitions>
        <sensor description="Logins" database="ForumDB"
            reading-max="1000" reading-min="0"
            polling-freq="3600"/>
        <sensor description="New Posts" database="ForumDB"
            reading-max="1000" reading-min="0"
            polling-freq="3600"/>
        <sensor description="New replies" database="ForumDB"
            reading-max="1000" reading-min="0"
            polling-freq="3600"/>
    </sensor-definitions>
    <condition-transformations>
        <transformation description="linear" class="LinearConditionTransformation"/>
    </condition-transformations>
    <sensor-conditions>
        <condition description="Login Activity Unusually Low"
            sensor="Logins"
            reading-max-heed="0" reading-min-heed="40"
            transformation="linear"
            range-samples="20" history-samples="24" forecast-samples="24"/>
        <condition description="Posting Activity Unusually Low"
            sensor="New Posts"
            reading-max-heed="0" reading-min-heed="40"
            transformation="linear"
            range-samples="20" history-samples="24" forecast-samples="24"/>
        <condition description="Reply Activity Unusually Low"
            sensor="New Replies"
            reading-max-heed="0" reading-min-heed="20"
            transformation="linear"
            range-samples="20" history-samples="24" forecast-samples="24"/>
    </sensor-conditions>
    <sensor-schema>
        <schema description="User Access Interference">
            <or>
             <paren>
                <and>
                    <schema-condition condition="Posting Activity Unusually Low"/>
                    <schema-condition condition="Reply Activity Unusually Low"/>
                </and>
                </paren>
                <schema-condition condition="Login Activity Unusually Low"/>
            </or>
        </schema>
    </sensor-schema>
    <display>
        <slider schema="User Access Interference" update-freq="3600"/>
    </display>
</gizmometer-config>
```

Figure 13: A sample schema and corresponding XML configuration file.

Perceiving

How a heed-based display facilitates perception of the heed is obviously important. The display should not require a significant amount of user attention to interpret and, contrary to an alarm, should minimize disruption. The display should clearly convey the heed and, if possible, support a forecast. The gizmometer display fulfills these design criteria by providing the user with two views which convey the current heed, the forecast, and a means of validating that forecast.

Elements of the Gizmometer Display

The Graph View

At its foundation, the gizmometer display is a graph of heed vs. time. The graph shows the history of heed values up to and including the current value, as well as a forecast of future values. The graph view of a single sensor condition is essentially a control chart where only the values between the upper and lower control lines are drawn.

The vertical axis of the graph view is always oriented with the heed value of 0 at the bottom and 1 at the top, even in situations where high sensor values indicate low heed. Thus, heed represents the need for attention, not the value of the measuring devices. Consistent orientation enables visual filtering or sorting, making it quick to spot the situation with the highest heed, or indeed if any situation warrants attention at all.

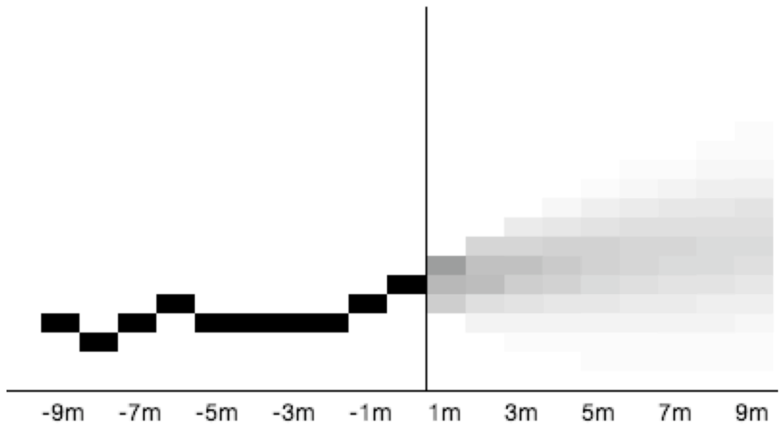


Figure 14: The Gizmometer graph view

Consistency is difficult to maintain for the horizontal axis, which is dependent upon the situation or sensor. The time scales will vary depending upon whether conditions change slowly or rapidly. This applies to the length of history displayed, as well as the length of forecast. Thus, although the vertical axis of heed need not contain units, the horizontal axis, time, must specify the display scale.

Resolution

In any implementation, sampling rate and resolution must be determined, both for heed value and for the time scale. Choosing a low resolution will help prevent a user from inferring accuracy in the graph that isn't there. Since the heed values themselves are not precise measurements, high resolution in the graph might cause a user to mistake precision for accuracy. In my tests of the heed system, heed was resolved with a precision of 0.05 units: in other words, there were only 20 values in the heed range between 0 and 1. Similarly, time was sampled so that only 20 units were rendered across the displayed domain: 10 units for history and 10 for forecast. These resolutions were chosen after tests with two users. Neither preferred variations of significantly different resolution than the 20 by 20. I caution any heed display designer to be wary of both too high of resolution as well as too low. High resolution conveys too much precision, and low resolution inhibits forecasting and early warning.

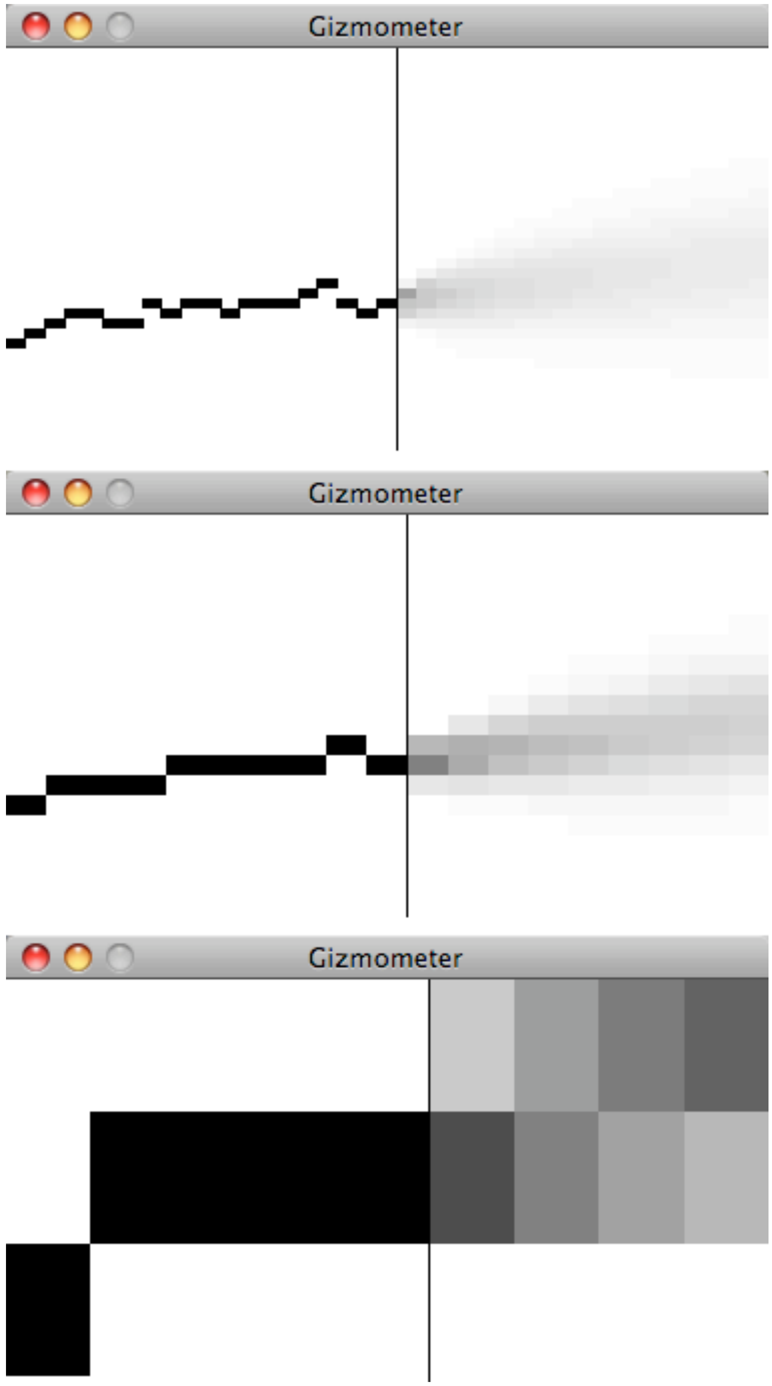


Figure 15: A comparison of three graph views rendered at different resolutions: top: 39 x 40, middle: 19 x 20, bottom: 9 x 3.

Mappings

The data in a gizmometer display are rendered in grayscale which is mapped to grid occupancy probability. Data from historical readings are rendered in black because the grid cells they occupy are certain. Forecast data is rendered in gray values where the level of gray indicates the probability of a grid cell in the graph view being occupied. For testing, I used a linear mapping between 0.0 and 1.0 for gray values ranging from 255 to 0: white to black. The result of the Kalman filter used to calculate the forecast lends itself well to this rendering scheme.

The Slider View

The gizmometer display employs a novel method of compressing the graph into a visualization that conveys the current heed value in context with historical and forecasted values, but without the time scale. The resulting display looks like a slider with two lines, one on either side of the slider. The slider has the same resolution as the range of the graph and is the same height. The current heed reading is represented by a circle at the same vertical position as the current reading would be represented in the graph; e.g. if the current reading is rendered at the tenth of twenty positions in the graph, the circle is rendered at the tenth of twenty positions on the slider. The circle appears and acts as a slider handle from more traditional slider interface elements; more on this later.

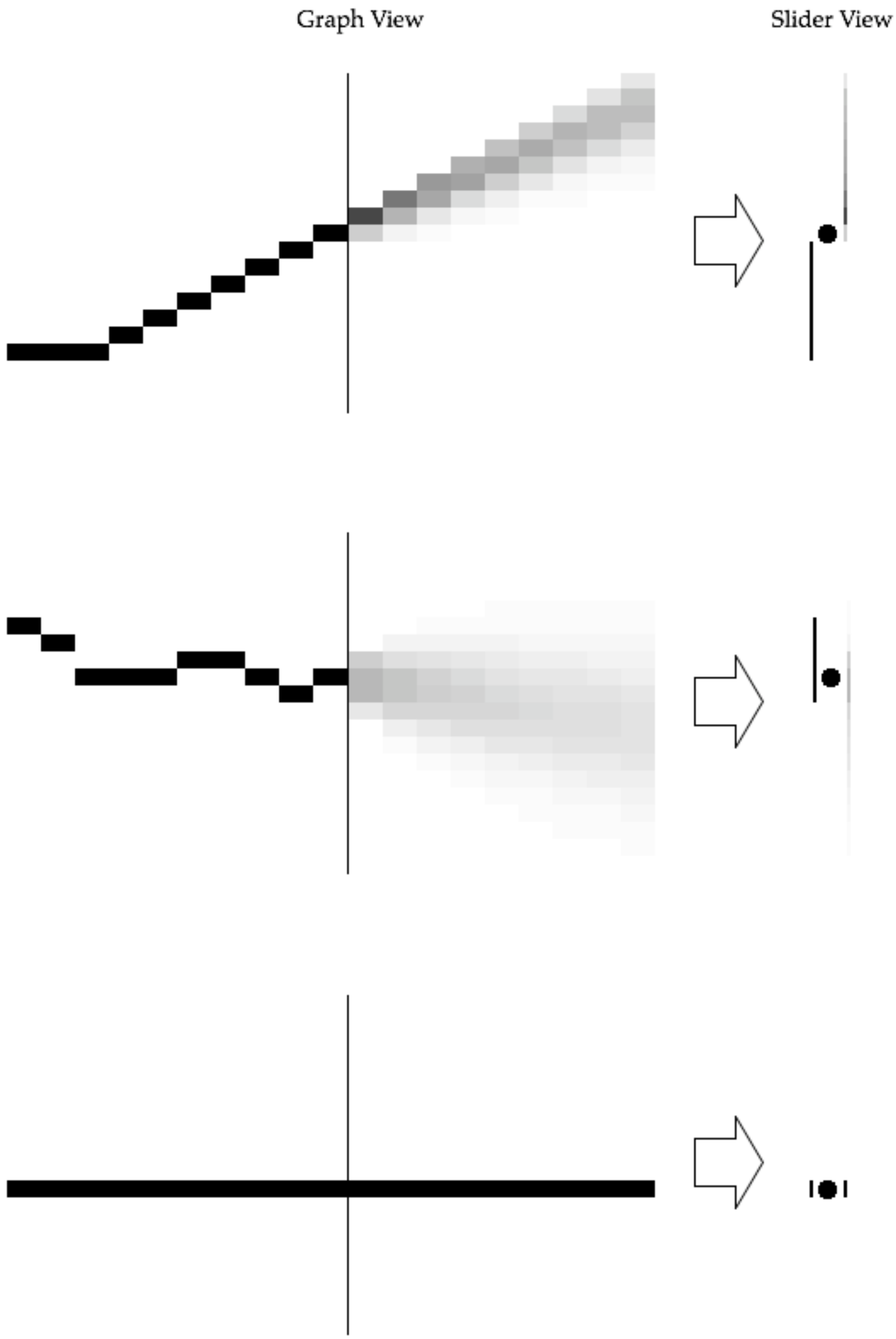


Figure 16: Three example slider views shown with their corresponding graph views.

Coverage

The vertical lines on either side of the slider provide context for the current reading by indicating the slider's history and forecast. These provide what has been called focus-plus-context in the interaction and visualization literatures (for a review of this technique, see (Leung and Apperley 1994). The focus of the display is the current value, and the context for that value is how it has changed recently and what it will likely change to soon. To convey this context, the lines identify the set of all positions the slider has been at, and will likely be, relying on the same grayscale mapping to probability as the graph display employs. I refer to the lines as "coverage" because I see them as representing which portions of the heed range are "covered" by the slider; either in the past or in the future. See Figure 16 for three different coverage examples.

The design of the slider display itself is inspired by the comic book renderings of objects in motion. Frequently, objects in motion are drawn with "action lines" to convey how quickly they are moving (McCloud 1993). The coverage lines are intended to evoke that same imagery, conveying to the observer not just the condition of the system, but how quickly and in what direction it is changing. Other techniques of drawing motion are potential candidates for creating heed-based interfaces; techniques such as arrows, forward lean, multiple stroboscopic images, or photographic blur (Cutting 2002).

I choose to employ motion lines because I can also use them to highlight their relationship to the underlying graph representation through an animated transformation. By retaining the same height and resolution of the original graph display, the focus-plus-context aspect of the slider display is enhanced with an animated transformation of the graph view into the slider view. The animation shows all points on the graph collapse horizontally toward the current reading; the historical points move to the right, and the forecast to the left. The animation ends when all the points have collapsed fully into the coverage lines. The animation helps to clarify how the coverage lines are to be read as well as making explicit the relationship between the slider display and the graph display. (In light of the interpretation of the coverage lines as action lines to depict the motion of the slider through time, the graph view of the gizmometer display could be interpreted as a different comics-inspired depiction of the slider's motion in time; namely as a multiple stroboscopic image.)

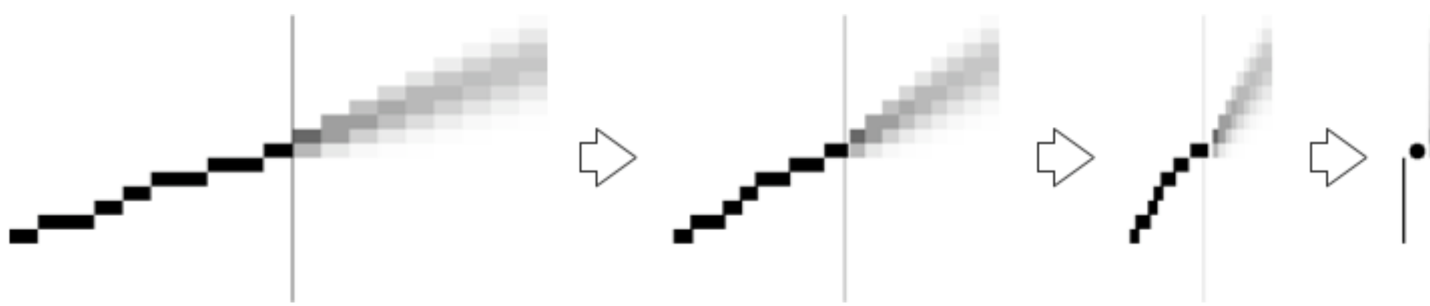


Figure 17: A graph view collapsing into a slider view.

Points on the graph have a position in time, a position in heed, and a grayscale value, while points in the coverage lines only have a position in heed and a grayscale value. How best to collapse the three former dimensions into the two latter involves another design decision. Three different heuristics lead to three different renderings:

1. Choose the highest grayscale value across the domain for each point in the range;
2. Calculate the mean grayscale value across the domain for each point in the range; or
3. Sum and normalize the grayscale value multiplied by the inverse distance from the origin.

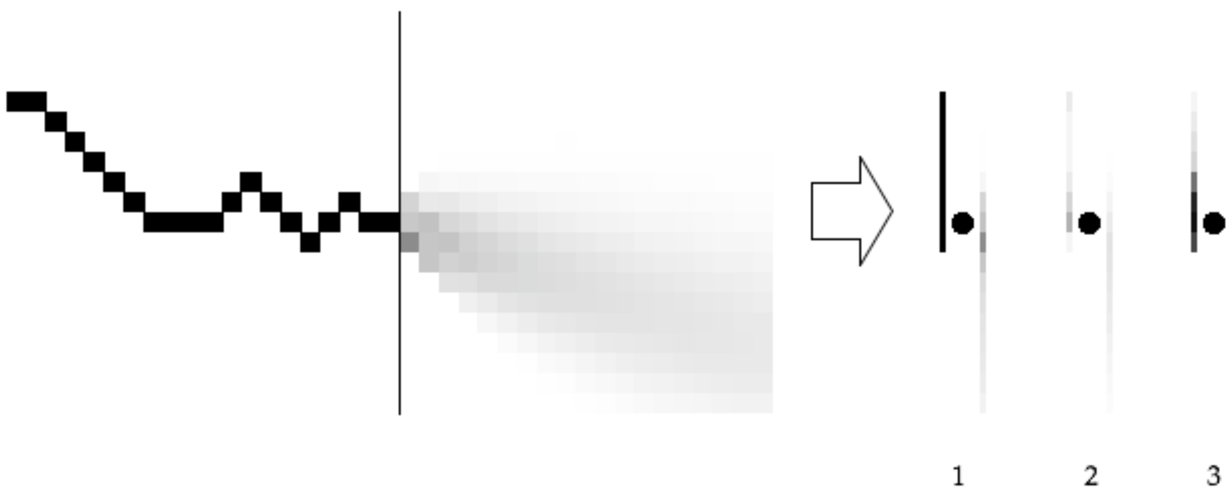


Figure 18: A visual comparison of three coverage rendering heuristics. 1: Electing the highest value in each row can obscure graph detail. 2: Using a frequency mapping instead of electing a value can result in a confusing forecast rendering. 3: Weighting the likelihood by age conveys detail but can confound the mapping of grayscale to likelihood.

1: The purpose of this heuristic is to retain the likelihood mapping from the graph display. By selecting the highest value, the coverage marks the history and the most likely forecast. However, this heuristic compromises by obscuring the detailed shape of the graph.

For example, a line that steadily rises from 0 to 0.5 and one that fluctuates between 0 and 0.5 would appear the same as a solid coverage line between 0 and 0.5.

2: The purpose of this heuristic is to better represent the detailed shape of the graph by changing the grayscale mapping from likelihood to occupancy frequency. The grayscale value now indicates how frequently the graph has a reading at each point in the range; the more frequently the graph had a reading at a particular point in the range, the closer to maximum the grayscale value at that point is. Thus, if the graph fluctuated over a portion of the range, that portion would appear darker than the rest of the range the graph covered.

Implementing this second heuristic is straightforward for historical data, but for the forecast data it would not result in a meaningful display. Depending on how the forecast was computed, many points in the forecast could be occupied, even though many of those points may have a low probability in the forecast. One way of modifying the mean occupancy heuristic is to multiply the occupancy with the likelihood. This would result in a dual mapping of likelihood and frequency, where the more frequently a point in the range is occupied, the darker that point is rendered in the coverage.

3: The purpose of this heuristic is to combine likelihood of occupancy with age. If a graph is long and covers a great deal of time, for many applications of heed, the behavior of the graph recently carries greater relevance than older data. One way of addressing this issue is by weighting the contribution of older data to the final grayscale value less than more recent data. This calculation is more complex that the other two, but the normalization step confounds the likelihood representation.

The Role of Color

All of the tested implementations of the gizmometer display are monochrome. Color could obviously be used as an additional dimension onto which another parameter could be mapped, but in the test interface all of the dimensions are already mapped to a position or grayscale value. However, color could be used as a redundant mapping. The display of Jam Factor (Auxer, Carroll et al. 2007), which bears some similarity to the gizmometer display, employs color in a redundant mapping. The Jam Factor display is intended to convey how congested a roadway is and consists of a slider without units and with extrema labeled "Clear" and "Jammed." The slider is colored in a gradient with green on the "Clear" end, red on the "Jammed" end, and yellow at the center. The display shows a box on the slider that is colored based on its position in the gradient and contains a number: the Jam Factor. The number is between 0 (Clear) and 10 (Jammed) and is displayed with one decimal place precision. In addition, small arrows are adjacent to the box and indicate the direction the Jam Factor value is changing and how quickly: one arrow means "slowly changing "and two or three arrows means "rapidly changing." In this Jam Factor display, color, slider position, and in fact, text, are mapped redundantly to the Jam Factor value. I have not found any testing of the display that discusses the value of the redundant mapping, but I cannot rule out its potential value. In the gizmometer display, it may be possible to map color to an otherwise already mapped parameter, such as heed value. Then, perhaps, higher heed values could color the slider red, and lower heed values could color it green, but, for now, I leave experimenting with such redundant mapping for future work.

Figure 19: Traffic.com's JamFactor display

The only use of color in the gizmometer display is in an optional element rendered at the top and bottom of each slider. These elements are small gradient icons, one in red on top and the other in blue or green at the bottom. These icons were necessary to help resolve a design conflict. The design conflict exists in the direction of reporting: as the situation becomes more important, the heed increases, and the slider moves up. However, the underlying data may be shifting numerically lower. This can occur, for example, when it is more important to attend a situation when fewer users log in, or when the temperature of an object drops too low, or when sales are flagging. The gradient icons were added as an optional reminder for new users of the interface; reminding them that even if a sensor value decreases, the graph is showing heed value that might very well increase with decreasing sensor values (for example, if the sensor measures free disk space or remaining fuel, low sensor values lead to increased heed). The top of the slider always represents high heed and the bottom low heed. During testing, some participants became confused by the apparent inversion of a graph they were familiar with and the reminder icons helped them acclimate to the new display. The color choice of a red-green mapping used for the reminder icons was based on a request from the participants only, and was not chosen based on any other criteria. Red-green is not a good choice for color pairs because of the prevalence of red-green color blindness in males: red-blue would be a superior choice, but one of my users insisted upon the use of red and green.

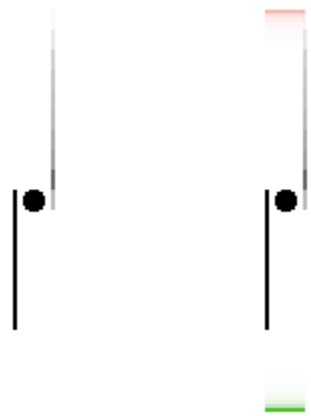


Figure 20: The slider view with and without reminder icons. Note: the choice of red and green color mapping was made by test subject.

Roll-over Labels

An aspect of the display most users immediately take note of when they first see the interface is the fact that individual sliders are not labeled. A label only appears when the mouse cursor moves over a slider. In effect, by not labeling the sliders, the interface creates a barrier to initiating an analysis task. This design choice come from the C-HIP model of warning discussed previously, and is intended to halt the processing of the interface at the attention stage and before the comprehension / memory stage. In other words, the barrier is intended to force the user to first identify if there is something that might need her attention soon, then to identify what that something is. If something does grab the user's attention, it forces an interruption in the form of required interaction, but if not, it would cause little to no disruption. This design choice was also made in an attempt to meet one of Endsley et al.'s design principles for situation awareness: "Minimize alarm disruptions to ongoing activities." This design choice was tested in subsequent example interfaces, but not fully validated.

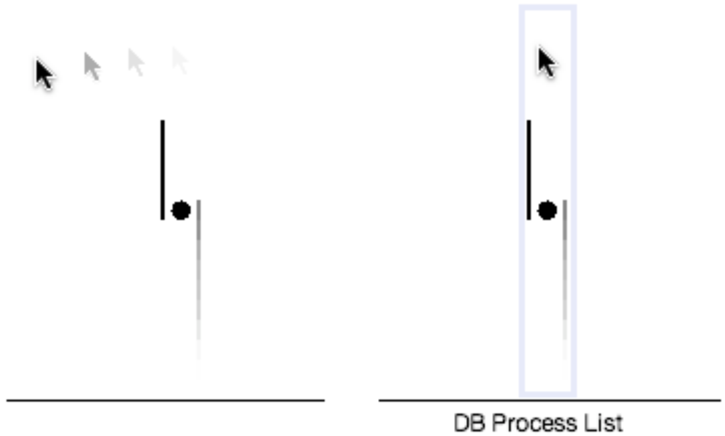


Figure 21: Roll-over labels.

Delving

If a situation appears to have a high heed, and if the observer wants to delve into the contributing factors that have led the display to that evaluation, the gizmometer interface supports several interactions that support clarifying the heed displayed. These include providing more detail than the slider view can provide by expanding into graph view, drilling-down into the schema to see just which factors are contributing to the final evaluation and how, and linking to other monitoring and analysis tools.

Expansion / Collapse

The slider view and the graph view can be toggled by clicking on the slider anywhere except directly over the position of the current reading (the slider handle). Following a click, the display shows the slider view expanding horizontally into the graph view, or the inverse animation of the graph view collapsing back into the slider view ( ). This interaction is meant to help the user acclimate to the slider display by highlighting the relationship of the two views: the familiar graph and the new slider representation. In addition, the graph view can be used as a first step in the analysis process by elaborating the rate and degree of change in heed over the recent history and to see when heed is forecast to reach high or low levels.

Drill-Down

As already discussed, many if not most, situations can be described by a combination of conditions. The XML hierarchy used to describe a situation in a gizmometer interface is a binary tree of leaf sensor conditions and logical operator nodes. The interface supports drilling down through this schema by traversing its binary tree representation top-down. With each subsequent right-click each gizmometer reveals the displays that show the heed of its leaves. By right-clicking on an individual gizmometer display, a new window containing a pair of gizmometer displays representing the next level down in the binary tree representation of the schema. Subsequently, in this fashion, the observer can reveal more and more of the tree, until the schema is fully revealed and only individual sensor conditions are left, which are the tree's leaves and cannot be expanded further. The drill-down interaction can be used by the observer to learn how each condition, or underlying combination of conditions, contributes to the final heed result, and in that way, help identify where to look to diagnose or analyze the situation.

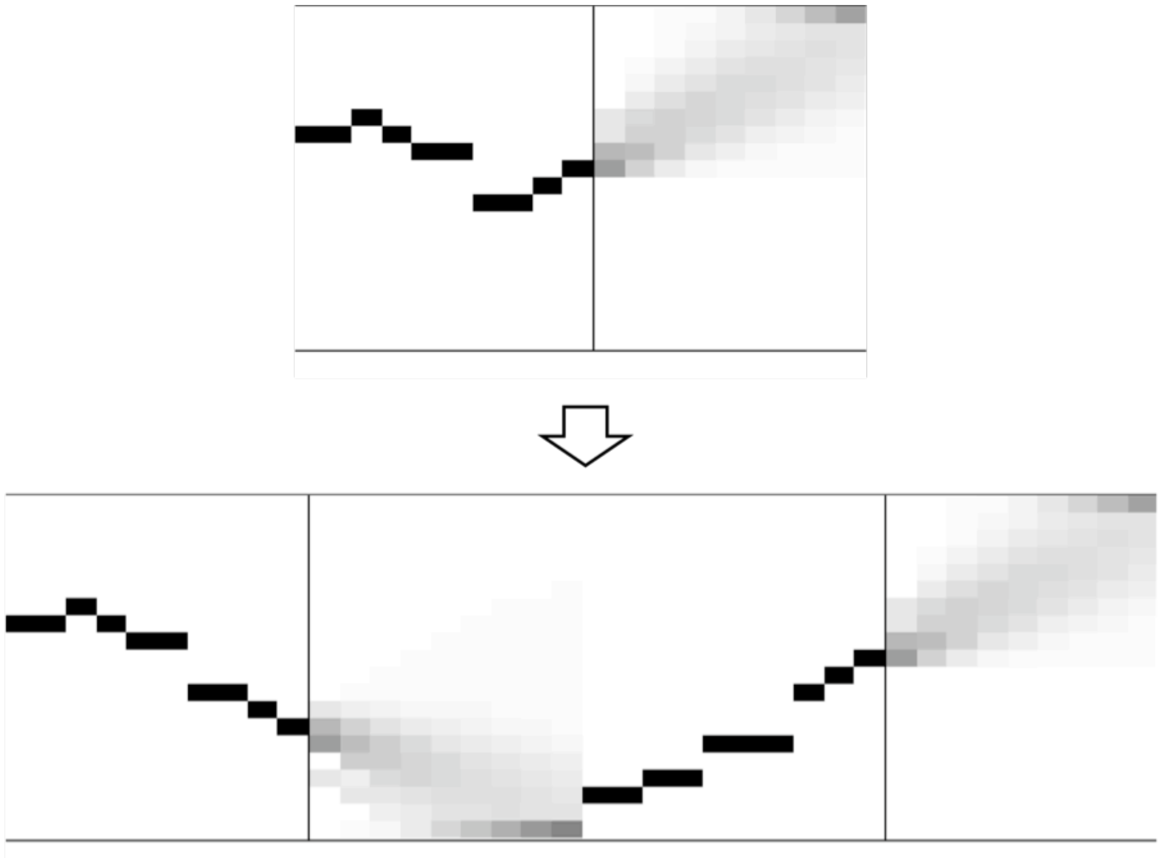


Figure 22: An example of drilling down into an OR-combination to reveal its constituents.

Linking

Any schema or sensor element may have a Uniform Resource Identifier associated with it. The Uniform Resource Identifier identifies an application or data source that can be launched by double-clicking on the label of a gizmometer. This allows for further analysis of the situation by bringing up an analysis tool or another appropriate monitoring tool.

Authoring: Condition Modification

If the observer, following some analysis, determines that the currently indicated heed value is inappropriate for the current situation, the interface facilitates the rapid modification of the underlying parameters. The indicated heed may be incorrect to the observer for two reasons. The first reason may be that the situation description is incomplete, meaning that it is missing either contributing factors or mitigating factors, or that it includes extraneous factors that obscure the heed of the situation. The second reason may be that the sensor conditions are inappropriate. The conditions may be too specific or too general, or no longer appropriate for the situation because the operating conditions, the observer's knowledge, or the system itself has changed. If the situation description is incomplete, it must be modified using a text editor and the situation description file discussed previously. Sensor conditions can be modified that way as well, but the gizmometer interface includes an additional novel method of directly modifying them as well.

In a traditional alarm-based interface, a user is granted only two forms of interaction: to activate an alarm or cancel one. In some circumstances, another interface may also allow the user to modify the alarm threshold. The gizmometer interface provides these same three interactions through a direct manipulation of the display itself.

Assuming an observer has investigated a situation and is confident that the heed indicated is inappropriate to the situation, then that means that the indicated heed is too high or too low. If the indicated heed is too high for the situation, reducing the indicated heed is much like canceling a traditional alarm. Likewise, if the indicated heed is too low for the situation, then the observer will want the indicated heed to be higher (at that moment and the next time the situation arises), which is essentially the same as activating a traditional alarm.

Finally, the observer may feel that the indicated heed is a little too high or a little too low, and may wish to refine the indicated heed. All of these interactions are accomplished in the same way: by dragging the slider handle to the position the observer thinks would be appropriate for the given situation (I will discuss this interaction in terms of manipulating the slider handle in the slider view, however, the interface allows for the same interaction in the graph view by dragging the indicated current reading). Since the interaction is accomplished by manipulating the informative display itself, it is an example of a direct manipulation interface. Direct manipulation interfaces have long been held as valuable (Hutchins, Hollan et al. 1985), (Frohlich 1993), however, implementing this interaction for a gizmometer poses two problems: how to determine which portion or portions of the schema to modify, and how to modify both boundaries by manipulating a one-dimensional interface element.

Figure 23: An example of adjusting sensor conditions by directly manipulating the gizmometer interface.

Choosing Which Portion of a Schema to Modify

Maria, our hypothetical systems administrator, must monitor the disk capacity of a 5-disk storage pool. Each disk has a capacity sensor, and for each, a sensor condition is defined with a minimum heed at 50% of capacity and maximum heed at 90% of capacity. Then the conditions are combined into a schema. If she combines the conditions with a series of ORs, she is saying that her attention might be needed if any of the disks nears capacity. (She could combine them with a series of ANDs, thereby saying that the more the entire pool of drives fills up, the more likely it is that she should turn her attention to the condition of the disks. She could also use AND-Ns and thereby say that she'd like to ignore the array until the last drive not near capacity starts to become full.) On one afternoon, drive 1 indicates 0 heed (41%), drive 2 indicated 0.5 heed (70%), drives 3 and 4 indicate 0.8 heed (82%), and drive 5 indicates 1.0 heed (93%).

Sensor Conditions: d1,d2,d3,d4,d5 : {m:50;M:90}%

Schema: ((((d1 OR d2) OR d3) OR d4) OR d5)

Data:

d1 = 41%, d2 = 70%, d3 = 82%, d4 = 82%, d5 = 93%

Heed:

$h(t_0)d1=0$, $h(t_0)d2=0.5$, $h(t_0)d3=0.8$, $h(t_0)d4=0.8$, $h(t_0)d5=1.0$

$h(t_0)$ =

(d1 OR d2) = 0.5

((d1 OR d2) OR d3) = 0.8

(((d1 OR d2) OR d3) OR d4) = 0.8

((((d1 OR d2) OR d3) OR d4) OR d5) = 1.0

Figure 24: An example situation, consisting of sensor conditions, sensor data, and the heed of a schema.

In this scenario, the highest heed of any sensor condition in the schema becomes the representative heed for the whole schema; in this case 1.0. Maria may look in on the situation by opening her disk analysis tools and see that disk 5 is at 93% of capacity. However, she also sees in the analysis tool that much of the disk is occupied by several temporary cache files which will be deleted soon, and that the largest files are not more than 10GB in size. This analysis tells her that, on her 250GB disks, she really never needs to look in on the drives until they're a lot closer to full. As such, for the current situation, she wishes to lower the indicated heed from 1.0 to 0.5, since she has determined that she probably could have afforded to continue to work on what she was doing before. If she clicks on the slider and drags it from the top to the middle, indicating that she wishes the gizmometer to indicate 0.5 heed, how can the application identify which aspect of the schema she wishes to adjust?

In this scenario, there are many combinations of parameter adjustments that will lead to the desired result. The use of the OR operator means that the highest value is propagated through the schema. As long as the heed values for all the sensor conditions that are above 0.5 (namely d3, d4, and d5) are at or below 0.5, and at least one is at 0.5, then the whole schema will indicate 0.5. Two of many possible combinations that would result in an indicated heed of 0.5 are seen in figure 25 and 26.

$h(t_0)d1=0$ *unmodified*

$h(t_0)d2=0.5$ *unmodified*

$h(t_0)d3=0.5$ *modified* from {m:50;M:90}% to {m:74;M:90}%

$h(t_0)d4=0.5$ *modified* from {m:50;M:90}% to {m:70;M:94}%

$h(t_0)d5=0.5$ *modified* from {m:50;M:90}% to {m:87;M:99}%

$h(t_0)$ =

(d1 OR d2) = 0.5

((d1 OR d2) OR d3) = 0.5

(((d1 OR d2) OR d3) OR d4) = 0.5

((((d1 OR d2) OR d3) OR d4) OR d5) = 0.5

Figure 25: One potential modification of the example situation.

$h(t_0)d1=0$ *unmodified*

$h(t_0)d2=0.3$ *modified* from {m:50;M:90}% to {m:60;M:80}%

$h(t_0)d3=0.3$ *modified* from {m:50;M:90}% to {m:76;M:94}%

$h(t_0)d4=0.3$ *modified* from {m:50;M:90}% to {m:76;M:94}%

$h(t_0)d5=0.5$ *modified* from {m:50;M:90}% to {m:88;M:98}%

$h(t_0)$ =

(d1 OR d2) = 0.3

((d1 OR d2) OR d3) = 0.3

(((d1 OR d2) OR d3) OR d4) = 0.3

((((d1 OR d2) OR d3) OR d4) OR d5) = 0.5

Figure 26: An alternate potential modification of the example situation.

This problem is an instance of the credit-assignment problem from cognitive science, wherein feedback only occurs after a complex set of actions making it difficult to identify which particular actions were responsible for the outcome. (Fu and Anderson 2008) (In a game of chess, for example, it is not easy to determine which moves led to defeat or victory.) This is still an unsolved problem, so in my implementation the user must explicitly make decisions and adjust each sensor condition individually.

Using the gizmometer interface Maria would drill down through the schema, identify the sensor conditions influencing the indicated heed of the whole schema, and then make individual adjustments to the sensor conditions. To do so, she would double-click on each display that indicates a high heed and thus drill down into the schema until the d5 sensor condition is revealed. Then she would click and drag it down to the midpoint of the slider, thereby requesting it read 0.5 heed. The slider would adjust, and then all the sliders above it in the schema's binary tree would adjust taking into account that the d5 now indicates 0.5 heed.

(d1 OR d2) = 0.5

((d1 OR d2) OR d3) = 0.8

(((d1 OR d2) OR d3) OR d4) = 0.8

((((d1 OR d2) OR d3) OR d4) OR d5) = 0.8

After the adjustment, the display for the entire schema would indicate a heed of 0.8, because even though d5 is 0.5 now, the 0.8 of d3 and of d4 supersede it in the OR operation. Maria might then choose to adjust the conditions of d3 and d4 after investigating their conditions in the disk analysis tool.

The process as described would be no different if the schema were a set of conditions joined by ANDs or AND-Ns. The credit-assignment problem is unchanged, only the method of calculation would change. Future work could improve the process by instantly drilling down to the conditions most influencing the result.

Modifying Two Parameters By Manipulating a One-Dimensional Interface Element

The second problem alluded to at the start of this section is also illustrated in the above example. Look again at the first adjustment example. d3 and d4 can have different sensor condition parameters ({m:74;M:90}% and {m:70;M:94}%), but still indicate the same heed 0.5. The interface only defines one degree of freedom: the resulting heed value, but the problem has two more degrees of freedom: the minimum, and the maximum sensor condition boundaries.

One way to deal with this discrepancy is to assert a heuristic that fixes one of the degrees of freedom. For example, if the adjustment is always made where the minimum is fixed, and only the maximum is adjusted, then the problem can be solved. In the above example, that would still present a problem. If the minimum were fixed for d5 and the requested heed was 0.5 when the disk was at 93% of capacity, then the maximum would have to be adjusted from 90 to 136%; but a disk can't be at more than 100% capacity. For this heuristic to work, it has to be elaborated with boundary conditions. Instead, the heuristic would become: fix the minimum, and adjust the maximum - if the maximum exceeds the sensor maximum, set the maximum to the sensor maximum, and adjust the minimum - if the minimum exceeds the sensor minimum, set the minimum to the sensor minimum, and adjust the requested heed value. Essentially, since the current reading is a fixed point, and the parameters must be moved, lowering the heed means to move the minimum closer to the reading and fixing the maximum, and raising the heed means to move the maximum closer and fixing the minimum.

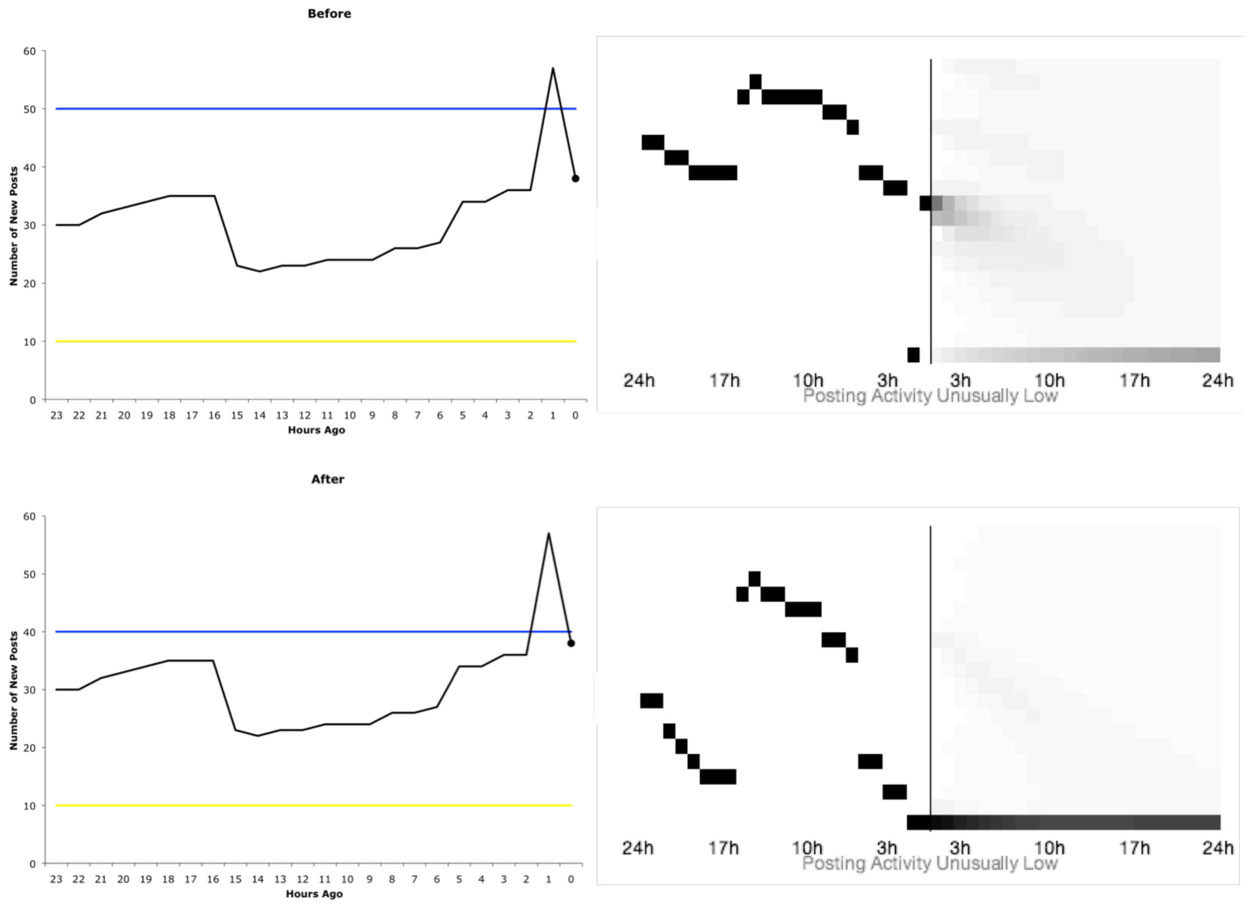


Figure 27: How adjusting the slider position modifies the ignore- and attend-boundaries of a sensor condition.

This rule, however, creates a problem in that adjustments are not directly reversible. If the user were to raise the indicated heed by 0.1, the maximum would shift slightly closer to the current reading. However, if then the user were to immediately lower the indicated heed by 0.1, the minimum would shift closer to the current reading instead of reversing the previous shift of the sensor condition maximum. To address this problem, either a separate "undo" operation is needed, or each new adjustment is made based on the original indicated heed value and a "finalize" operation is needed to confirm the adjustment.

The method as described so far results in the iterative narrowing of the range between sensor condition boundaries with each adjustment. To expand the range either an action that resets the minimum to the sensor minimum or resets the maximum to the sensor maximum is needed, or an action that triggers a reversal of the narrowing heuristic into a widening one is needed. When a heed value of 0.0 is requested, and the adjustment results in the sensor condition minimum being shifted to the current sensor reading, then the maximum can be left as it is, or it could be shifted to any other value, including the sensor maximum. Likewise, when a heed of 1.0 is requested and the maximum is shifted to the current sensor reading, the minimum could be shifted all the way back to the sensor minimum. Note: if the sensor condition minimum is numerically greater than its maximum, the expansion shift must retain that ordering.

The range will also expand when the current reading is outside the sensor condition range, and any adjustment is made. For example, if the range is between 50% and 80% for a sensor that reads 40%, and the user requests that the heed be raised from 0.0 to 0.5, then the range widens to 10% and 80%. This is because the heuristic keeps the maximum fixed when the heed is raised, and to accommodate the current reading of 40%, the minimum must be lowered to 10%.

The gizmometer interface is one example of an interface built around heed. It supports the authorship of the situations that the observer wishes to monitor, as well as supports identifying if a situation may need the observer's attention. Finally, it supports the initiation of an analysis of those situations, and eventual modification of the situation's parameters.

Chapter 5: Testing Heed-based Displays

The Test Subjects

Four subjects, working in different jobs with significantly different duties, were studied to explore the ability of heed to capture a situation that each monitors, and to learn if the gizmometer interface is an effective tool for increasing situational awareness. Subjects were studied in their usual work environments. They were observed and interviewed to ascertain the degree which the test interfaces affected their situation awareness: in particular, which level of SA was affected and how. The three levels of SA are: perception (level 1), comprehension (level 2), and projection (level 3). An increase in level 1 SA would be indicated by an increased awareness of individual parameters relevant to the testing situations. An increase in level 2 SA would be indicated by an increased awareness of the state of a situation as a whole, and an increase in level 3 SA would be indicated by an increased awareness of how the state of the situation will change in the near future.

Each subject has unique responsibilities in their respective work environments. Subject A is the Chief Operations Officer in a small startup company. In addition to his executive duties, his responsibilities include the purchasing, configuration, and maintenance of the company's servers and desktops, as well as being lead developer of the company's website software. Subject B is also an executive in a small startup company. However, she is not responsible for any technical aspects of the company's operation; instead, she is responsible for the financial well-being of the company. Subject C is responsible for the design, creation, maintenance, and discussion moderation of a community-oriented website. Like subject A, subject C is a lead developer for his website, but his organization utilizes co-located servers, and thus relies on the co-location service to administrate the hardware. Subject D is the chief veterinarian at a very large animal shelter. In addition to his duties administering care as a veterinarian, he is responsible for the operations of the clinic at the shelter (which include budgetary as well as medical concerns), the welfare of the animals at the shelter, and consultations in criminal cases.

Subject A: Server Condition Monitoring

Work Situation And Issues

In addition to his executive duties, subject A is the lead developer and the sole systems administrator for dozens of desktop computers, a mail server, a database server, and a web server. The number of duties, their importance to the operation of the company, and their complex nature means that the time he has available to dedicate to the monitoring of his servers is very limited. Studying A's monitoring methods revealed that, despite having powerful monitoring tools that display many graphs of server conditions and many email alerts sent to him on the state of his servers, he would most often rely on weak, but quick to use, monitoring tools such as the UNIX uptime command or the MySQL process list command to inform him of the condition of his servers. Compounding the issue, we learned that he only uses that method when he happens to be logged in to a server via the terminal, which was only a few times a day. As a result, his awareness of the condition of his servers comes largely from infrequent sampling of processor load.

Due to the subject's method of monitoring, he can be caught off-guard when a user cannot get access to the site and calls the support line. Should that happen, he must turn his full attention to monitoring at the cost of all his other duties in order to determine the cause of the problem. In fact, the subject recounted an incident in which he and his entire development team wasted a day closely monitoring the servers to identify and diagnose a problem that only arose after the application was deployed. This subject's lack of available time to fully engage in monitoring activities, the rapidly changing nature of server conditions, and the fact that situations that need his attention arise infrequently, makes him a good candidate for an easy to use monitoring tool.

Design

The goal of the interface in this study is to increase awareness in the user with regard to the condition of his servers and to the situations that need his attention without adding to his workload. The monitoring situation that we studied is one wherein the services provided by his servers are being heavily taxed — either because users are accessing them in large numbers, or because there is a bug in the site's application code. Situations wherein the hardware fails or services are compromised due to an intrusion were excluded from this test because the methods the subject employs to monitor for those situations were not identified in the initial study.

He monitors for server-taxing situations by sampling the processor load averages on the web and mail servers, and the length of the process list on the database server when he has opportunity to do so. To support this monitoring task, we authored the situation description based on this method of monitoring. The interface created for testing with the subject is a set of three gizmometers: two based on the 1-minute load average of the web server and of the mail server, and one based on the length of the process list on the database server. The initial thresholds for each gizmometer were based on the values gathered from interviews with the subject based on his experience with the output of the uptime and process list commands. The final display is shown in Figure 28, and indicates high heed when processor load averages are high or when the process list is long.

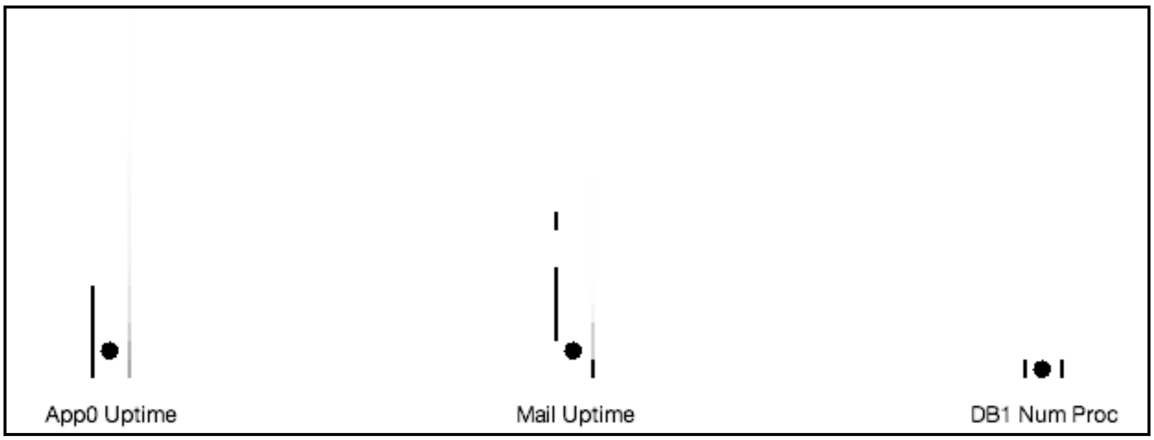


Figure 28: The test display for subject A.

Deployment And Test

The interface was implemented using Java and the Processing.org libraries, and a series of Perl and shell scripts collected data. The three servers were polled every half-second and the samples stored in a MySQL database on the test machine (a Pentium III 866 MHz tablet pc with 256MB of memory running Microsoft Windows XP). A medium term test was set up using this equipment wherein the application was left running on the desk of the user for one week. Throughout that week, the user was observed twice, once at the start of the week, and again at the end of the week. During the week, the user was interviewed twice more by telephone.

The subject reported a decrease in his overall concern for the condition of his servers throughout the week. In other words, the heed display was able to increase his confidence in the condition of the servers, and because he did not make significant changes to the servers during the testing period, the increase of confidence can be attributed to an increase of awareness. The awareness in this case was not due to additional specific knowledge, but about the state of the servers in general, suggesting that the increase of awareness occurred at level 2. During the first day of testing, the subject learned that his estimates for what constituted high load on his servers had been wrong. By adjusting the thresholds to suit his servers better during that initialization phase, he improved his understanding of the conditions under which his servers typically operate. This result indicates an increase in level 1 SA. Additionally, the subject reported that one of his staff discovered an erroneous database job after seeing the test display. The staff member had accidentally left a large job running on the database server; the display indicated a high heed for the database process list, which alerted him to the situation. This incident suggests that the display was easy to read and interpret, because the staff member was not part of the test, and had only been casually instructed on its use by the test subject. Furthermore, this incident is an additional example of a person increasing their level 2 SA as a result of interaction with the display because, here, the staff member exhibited a direct association of a heed value with a situation underway on one of the servers.

Subject B: Report Monitoring

Work Situation and Issues

As an executive, subject B's duties require her to maintain an awareness of many aspects of her company: primarily financial conditions that affect her company's revenue, sales, and expenses. The subject has many powerful monitoring tools to aid her, including daily, weekly, and monthly reports covering the financial health of her company. Unfortunately she has so many duties and frequent, urgent situations that require her immediate attention that she easily forgets to read the reports. As a result, the subject has only a limited awareness of the performance of all aspects of her business at any given moment. During an interview, the subject described how this became evident to her in an incident wherein she was shocked to learn that a particular revenue-generating service was not only seriously underperforming, but had been so for months. The subject had not read the reports for that program in such a long time that the she had lost her awareness of that situation. Further questioning revealed that she is apt to put off reading the reports because they change infrequently, giving her the impression that the situation will not change. The program had performed adequately at first, with each report reading similarly to the one before. This resulted in giving B a false sense of security without awareness for the situation. B's limited time and infrequently changing but important situations make her a good candidate for an easy-to-use monitoring tool.

Design

The goal of the interface in this study is to remind the subject to read one of her reports more regularly and thereby increase her awareness of the program detailed in that report, all without creating too much interference in her working environment. When reading the report, the situation that she monitors for is one wherein the program is performing unexpectedly. The daily reports present sales program utilization data for four clients in that program. The report details the total utilization, daily change in utilization, and the daily deviation from expected utilization. Based on interviews with the subject, she reads the reports looking for spikes in the daily change, and looking at how well they are meeting their expected utilization goals. The subject also estimated that ideally, she would read the report weekly, and at the very least, every other week.

To study this reminder task, an interface was created consisting of a single gizmometer, that indicates the heed of the situation as a whole: in other words, heed indicates the importance of reading the report. The heed is based on the time since the report was last read, which is captured from the web page that generates the report. The more time that has elapsed since the report was read, the more important it is to read it. The heed is also based on the content of the report itself, which details a business program's utilization. The greater the spike or dip in utilization, the more important it is to read it. Also, the farther utilization is from expected performance, the more important it is to read the report. The subject can expand the single gizmometer at any time to reveal the heed associated with any of the constituent conditions that influence the situation. The initial thresholds for each of these sensor conditions were set using values reported by the subject based on her experience with the program and what was recorded in the report at the time of the test.

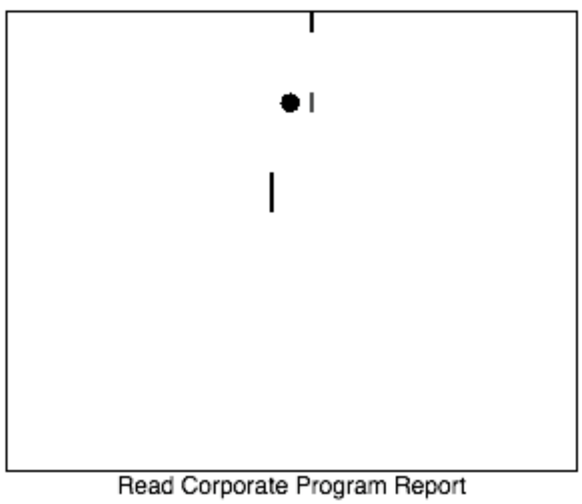


Figure 29: The test display for subject B. This figure is elaborated in Appendix B.

Deployment and Test

The interface was implemented using Java and the Processing.org libraries, and a series of Perl and PHP scripts collected data. The scripts polled the company database every day and recorded samples to a MySQL database on the test machine, which was a PowerPC G4 667 MHz laptop with 768MB of memory running Macintosh OS X 10.4. A medium term test was set up using this equipment wherein the application was left running on a side table four feet to the right and behind the desk out of constant direct sight of the subject for two weeks. After each week, the user was interviewed.

The primary goal of this application was to remind the subject to read her reports when necessary. Unfortunately, the interface failed to spur the subject to read her report during the two-week test. The subject stated in her final interview that she had been reminded to read her report when she glanced at the display as she left the office that day, however, this result is inconclusive. The failure in this case may be attributed, in part, to an error in sensor data. An accidental application restart during the two-week session prevented the heed associated with the age of the report from ever becoming high.

The secondary goal of this application was to increase the subject's awareness for the program detailed in the report. The anticipated outcome of the study was that an increase of awareness would come from the subject reading the report. However, much like subject A, subject B reported a decrease in her level of concern for the program over the course of the test and an increase in her level of awareness for the performance of the program. Although the subject did not indicate any increase in her level 1 SA, as indicated by her lack of any specific knowledge of program's performance, she appeared to establish level 2 SA because she developed a confident evaluation of the program as a whole despite underperformance of that program throughout the test period. This result is bolstered by the subject's response to the test: her stress with regard to the program decreased sufficiently that she asked for additional interfaces to monitor other aspects of the business that were significant sources of stress to her. The interface appears to have established in the subject an awareness for the situation, though it did not spur the action it was intended to.

Subject C: User Experience Monitoring

Work Situation and Issues

Subject C is in charge of a website devoted to the support of a large non-profit community. He designs and develops software including a discussion forum, which he must moderate as well. Although his role as developer supersedes his role as moderator, subject C considers it important that his users have a positive experience when interacting with one another in the user forum. However, the subject has difficulty monitoring for situations in the forum he must moderate because he does not have any tools designed for the monitoring of online discussion. To identify situations that may require moderation, the subject will occasionally skim through the forums looking for evidence of certain participant behavior that he believes may lead to a negative experience for the rest of the community. In particular, the subject describes looking for conversation that is dominated by only a few users, or for forums in which conversation never seems to materialize because too many users are starting threads to which no one responds. This method of assessment is very time-consuming and frequently yields nothing of concern, and as a result, the subject performs it rarely. Furthermore, as the site's developer, he must frequently make changes to the software, and may accidentally introduce bugs. The subject recounted that on at least one occasion, he introduced a bug that prevented a subset of his users from accessing the site; a fact he only learned of days later. An inability to access the site will obviously impact his users' experience negatively as well. The subject is responsible for application-level monitoring, but since he relies on co-location of his servers, the monitoring and maintenance of the hardware is left to his service provider. The dearth of available tools to support monitoring for these situations, the rarity of the situations, and the time-consuming and evaluative nature of their assessment make this subject an excellent candidate for an easy-to-use, configurable monitoring tool.

Design

The goal of the interface in this study is to increase awareness of a negative user experience with his site. We studied two of the situations that he monitors: ones wherein an application-level problem prevents users from accessing or contributing to the site, and ones wherein forum conversation becomes unbalanced. To capture the data relevant to such situations, sensors were integrated into his website's code to sample the numbers of posts made in the forum, the numbers of replies posted, and the number of threads that have been started to which no user has replied. The subject described his method of evaluating conversation as becoming unbalanced when there is too much or too little conversation or when there are too many posts to which no one replies. The more threads without replies are created each day, or the more posts or replies are created in a day, the more important it becomes that he make time to moderate the forums. The interface was configured to convey that these conditions may have occurred by defining a sensor condition for these three conditions and then combining them with OR operators.

Detecting that a bug has prevented users from accessing the site is based on a significant drop in the numbers of people logging in, posting, or replying to posts. The greater the drop, the more important it becomes that he turn his attention to investigating for a problem and the higher the heed is for the situation. A second gizmometer was configured to support monitoring for this situation. As with the other situation in this study, the configuration in this case was also an OR combination of the relevant sensor conditions.

The final display in this study is a gizmometer for each of the two situations. In this case, the user had no specific knowledge of what the thresholds for these situations should be, so the initial thresholds were estimated by the user and a tuning phase was employed during initial testing to refine those estimates.

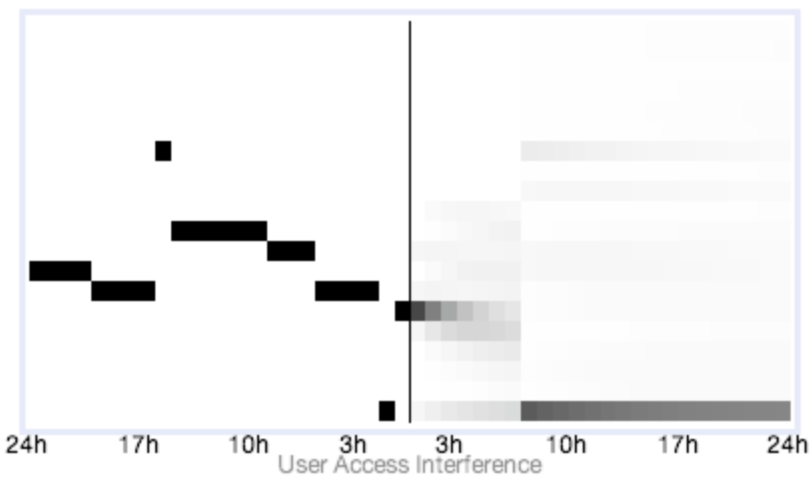


Figure 30: The test display for subject C.

Deployment and Test

The interface was left running for a week with the participant. The interface again was implemented using Java and the Processing.org libraries, and a series of Perl and shell scripts collected data. The scripts would poll the Ruby on Rails application database that constitutes the subject's website and write the current readings in to a MySQL database on the test machine, which was a PowerPC 867 MHz Apple PowerBook with 1GB of memory running Macintosh OS X 10.4. The equipment was left running on a table four feet to the right of the participant's desk. Throughout that week, the user was observed twice, once at the start of the week, and again at the end of the week. During the week, the user was interviewed once more.

The goal of the interface was increased awareness of his users' experience with the subject's website. This appeared to have occurred, but not as anticipated. Subject C frequently modified the thresholds associated with each situation throughout the test period. He explained that he made these modifications to make the display match his existing awareness of his users' experience and the site's functioning. The subject initially configured the thresholds based on his best estimates. During the testing period, however, the subject made changes to the condition thresholds when he felt the display did not match his awareness of the situation. Although there was no evidence to suggest that the modifications were performed as a result of any deliberate monitoring activity to confirm his awareness, the subject explained that prior to making any changes he had spent a significant amount of time working on the application code or otherwise involved with the site and its activity, and therefore felt confident in his awareness. As a result of this readjustment activity, the subject learned that his original understanding for how his users utilized the site had been incorrect and that following the test his understanding had been improved. This suggests an increase in the subject's level 1 SA, but unexpectedly, through a process of making the display match his level 2 SA.

The subject's experience with the display had an additional unexpected result; namely, that he went on to modify the display's configuration on his own to monitor for entirely new situations. A follow-up interview revealed that the subject had found the display a powerful learning tool. He described his experience with the display as revealing where his intuition had failed him, and sought to utilize the display to test his intuition for other aspects of the site's utilization. The subject's own display consisted of twenty-four conditions related to user activity, such as content tagging, private messaging activity, as well as social-networking activities such as testimonial posting, and other relationship identification activities. As with the first test, the subject initially defined thresholds for each of these activities, and used the display to monitor when his estimates had been incorrect. The fact that the subject felt compelled to create his own display and utilize it as a tool for testing and learning further supports the effectiveness of the interface to help increase level 1 SA.

Subject D: Disease Outbreak Monitoring

Work Situation And Issues

The animal shelter at which subject D is the chief veterinarian occasionally has outbreaks of infectious disease among the animal population. Such outbreaks are of grave concern to him because an outbreak can result in the loss of animal life as well as provoking significant unforeseen expenses in the care of the sick animals. Containing an outbreak is best accomplished through isolation. Ideally, once an animal has been identified as infected, it is moved to an isolation room, but an isolation location is not always available. Even if the infected animal is moved, each illness has an incubation period of up to two weeks during which an animal is still contagious. It is possible that other animals with which the sick animal was housed before being isolated may yet become ill. The long delay in symptom presentation, combined with frequent animal contact and compounded by a necessity to move animals to different rooms at the shelter on occasion, makes determining the potential for an outbreak a complex problem that is difficult to monitor for. The shelter has some means of combating an outbreak, but these are expensive and time-consuming. In particular, the subject can call for a room to be aggressively disinfected once an infected animal has been removed from it. Before calling for such action, he must be certain that all the animals in that room are free from infection, and therefore must wait until the gestation period has expired. To be certain, the subject must consult the medical records of each the animals in that room, as well as the records of the animals that have recently been in contact with those animals. This is an extremely time-consuming activity for the subject, one for which he rarely has time. Finally, this situation is further complicated for the subject by the distribution of information relevant to maintaining an awareness of the situation. It is typically one of many volunteers who identifies an animal as potentially ill and brings it to the clinic, where one of several veterinarians will diagnose the animal, which if ill, is moved to a third location for isolation and care. As a result, the subject's awareness of how many animals have been treated for an illness, when each incident occurred, and therefore if and when outbreak mitigation procedures are necessary, is limited. The subject's limited available time, the complexity of making an assessment, and the lack of any system to support outbreak potential awareness makes this user an excellent candidate for a monitoring tool.

Design

The goal of this interface is to indicate when it is important to investigate if outbreak mitigation procedures need to be implemented. The conditions influencing that situation are based on the how many animals have become ill recently, and in what room they were housed prior to their diagnosis. The more animals that have been diagnosed recently in a room, the greater the heed is for that room. The more rooms that indicate a high heed, the more important it is to investigate if outbreak containment procedures are necessary.

To accomplish this, an interface was built to display the heed associated with each room. The heed for each room is based on the last time an animal was diagnosed with an illness in that room, and the number of animals from that room that have been diagnosed. The data for this application was collected from historical records kept in a database by the shelter. The data collected for the application consist of logs of animal locations and dates of disease diagnosis.

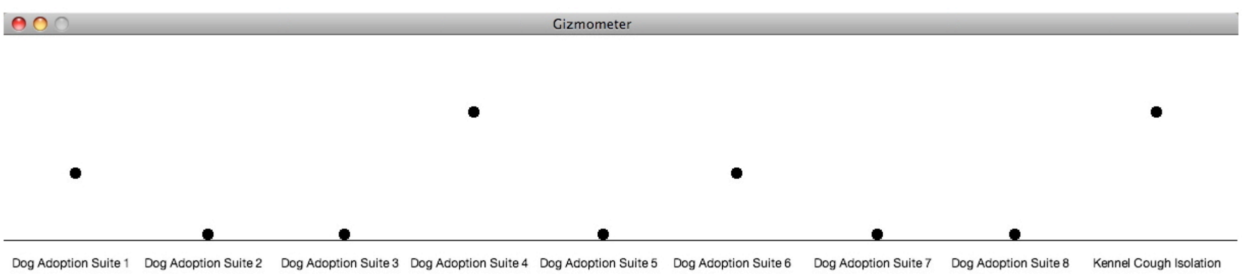


Figure 31: The test display for subject D.

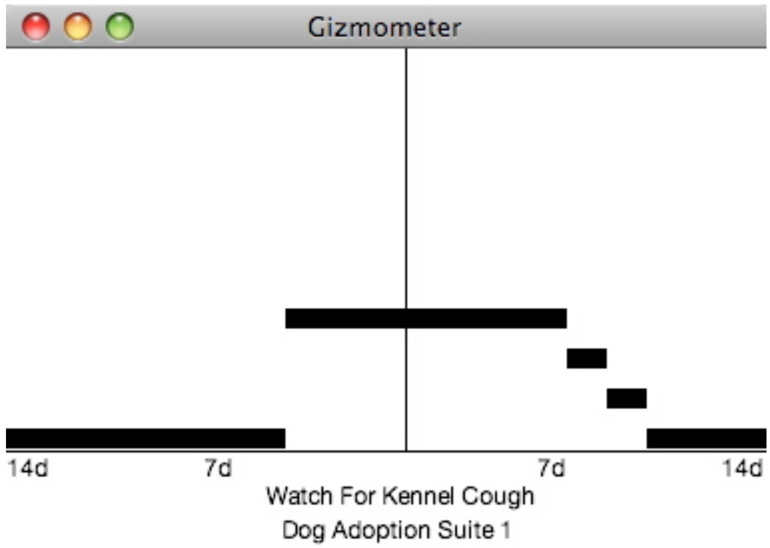


Figure 32: The heed of a single incident for the interface built for subject D.

The gestation period for the illness being monitored for, in this case kennel cough, is about ten days. Because the chance of the infection diminishes over time, the heed associated with each incident diminishes with the age of the incident. Therefore the forecast for heed could be a simple function of the age of an incident; the older an incident is, the lower its heed will be. As opposed to the previous three interfaces, where a Kalman filter was used to forecast heed, in this application the forecast for heed could be calculated more precisely. Figure 32 shows the heed associated with one incident.

Deployment And Test

The interface for this application could not be tested in real-time because the underlying data could not be collected without manual intervention. Instead, a display was built using the historical information, and the interface was presented as a prototype to the subject while being interviewed. The subject was presented with the interface as two series of screenshots: one indicating the heed associated with a single room, and a second showing the heed for nine rooms. The goal of this interface was to increase the subject's awareness of a potential outbreak and the need to investigate that situation. Without a live test spanning days or weeks, an increase of awareness could not be studied. However, the subject responded to the prototype immediately, interpreting the situation and identifying the possibility of an outbreak based on the display. The subject responded with an additional interpretation, suggesting that the same interface could support two additional needs of the clinic. The subject believed that publishing the interface to donors could keep them informed of outbreak occurrences and the clinic's track record for coping with outbreaks. He also believed that the interface could aid in mitigating an outbreak before it occurs. The subject interpreted the heed for each room as the danger of that room for spreading an infection. He suggested that the display be made available to the volunteers working at the shelter to aid them in making decisions about where to house newly admitted animals or where to put animals that needed to be moved from their current room. These two interpretations suggest that the subject believed that the interface could communicate level 2 SA to the donors, and level 1 SA to the volunteers, which itself suggests that the subject could interpret information relevant to his own level 1 and 2 SA. These results are inconclusive, but encourage further testing of this interface methodology in situations with many stakeholders and where conditions influencing those situations are not purely technological in nature.

Summary of Results

The goal of each interface was to provide the subjects with an easy-to-use tool to increase their awareness of a situation important to them. It is difficult to assess the impact of the display on SA because impact is increased awareness The display appears to be successful at increasing awareness, in particular at increasing level 2 SA. Subjects A and B appeared to have increased their level 2 SA as indicated by an apparent decrease in their reported level of stress with regard to the situations the display was monitoring. Subject D, though only presented with a prototype, recognized how the interface could increase his level 2 SA, as indicated by his recognition of how the interface could similarly affect the SA of other stakeholders in the situation being monitored for. Subject C did not increase his level 2 SA, but did treat the interface as a tool to capture his level 2 SA, and thereby increase his level 1 SA. In the case of subject C we have evidence that the interface facilitated the establishment of level 1 SA for situations that the subject defined himself. This finding suggests that a heed-based display can be used as an interface for learning as well as monitoring.

It was important that the interface be easy to use as well, and not interfere with the subject's work or add to their workload. All subjects understood how to read and interpret the display very quickly, needing little if any follow-up instruction. In fact, the result of testing with subject A revealed that even people receiving no formal instruction could read the display. However, testing revealed a design conflict in the use of the interface. The conflict in the direction of reporting was a challenge, particularly for subject C. The mapping of high heed to low logins or other low parameters was a source of confusion. The conflict was successfully resolved with the addition of icons to the interface to act as reminders of the direction of mapping. None of the subjects reported spending a significant amount of time interacting with the display, and during the observation sessions with subjects A, B, and C, we observed little more than cursory glances at the display once the interface was initialized. It appears that the interface was able to convey the situation with little interaction, suggesting that the interface is well-suited for use as an ambient interface. However, these studies also found that the interface was unsuccessful at spurring follow-up action. Subjects did not report, nor were any incidents observed, in which, after seeing the display, the subject investigated a situation. The incident reported by subject A in which another person took action after seeing the display and the report by subject B that she was reminded to read her report as she left the office are the closest such activity, but are not directly supportive of the overall goal of a heed display to support monitoring activities. The situations in each of these studies occur rarely in situ, and did not occur during our testing sessions. If each test had spanned a longer period of time, this result might have been different. Testing under more controlled circumstances to better identify the interface's ability to spur investigative action is planned for future work.

The four test cases also exemplify how a heed-based display can be applied where a traditional alarm, dashboard, or report could not have been. Subject A already had alarms that arrived in his email inbox and were ignored, and had dashboards which were sufficiently cumbersome that he resorted to occasional sampling of processor load to develop level 1 SA, which the initialization of the test interface demonstrated had been erroneous. The existing tools available to the subject did not provide him with the means necessary to maintain his awareness for the system. Instead, the subject relied on his frequent interaction with the system to maintain awareness. By frequently modifying the application code, performing maintenance tasks, and acting as a user of the application, the subject developed some degree of level 2 SA. However, the subject's report of reduced stress with regard to the system during the test suggests that there was room for his comprehension of the system to be improved. An alarm could not have helped him do that without constant interruption. Because the system condition changes second by second, a report would not have been able to help either, since it would not be generated in a timely manner, and would also have to interrupt the subject frequently if he were to read each report (which we learned from subject B can easily be put off as well). A dashboard could provide subject A with the information necessary, but to make it less cumbersome, it must be configurable; enabling the subject to specify which system parameters are displayed. The subject described the interaction with his dashboards as something that he skimmed looking for problems. This means that he did not use his dashboards to maintain awareness, but rather to sample for outliers. The application of heed in this case enables the subject to monitor with greater frequency than with a report or an alarm, and enables the establishment of awareness by enabling the subject to limit the scope of information displayed in dashboard-like manner. This conclusion supports both the Endsley et al.'s "complexity design principles" and Hollnagel and Woods "designing for complexity" guidelines.

A dashboard-like display must be configurable by the user. We see from the test with subject C that the rapid and simple modification can enable the establishment of level 1 SA. Like subject A, subject C developed his level 2 SA through frequent contact with the system being monitored. However, the subject learned that what he thought he knew about the usage of the system was wrong. By enabling both a means of authorship and a simple method of adjustment, the subject was able to define what he wanted to learn about, and then support refinement of that knowledge through monitoring and readjustment.

The situations Subject B needed to monitor were in her reports, but this does no good if she does not read the reports; the longer it has been since she reads any report, the lower her SA is, and the less likely it is that anything will remind her to read a report. Furthermore, applying an alarm to the situation in order to remind her to read the report, not only fails to increase her SA, but it is a source of disruption. A heed-based display minimizes disruption, but maintains a continuous reminder by increasing the heed of the situation based on how stale the knowledge of the observer is (e.g., how long it has been since the report was last read).

No monitoring tools exist to support subject D. Even though the underlying information such as animal diagnoses and housing information can be presented as a report, there is so much information that must be synthesized to make an assessment that a report would be very time-consuming to produce. The combination of evaluative assessment, complexity of information, rarity of occurrence, and importance of vigilance means that a report would be insufficient. An alarm or a dashboard would be particularly difficult to implement, but a heed-based display was not only possible to build, the subject identified additional applications for the display associated with the same situation immediately. From this we learned that a heed-based display can be applied to a situation that is not technological in nature, and could support situation awareness in a multi-operator or multi-stakeholder environment.

Summary And Overview

The most important challenges are in capturing the situations being monitored. I found that people look at monitoring in terms of either conditions or outcomes. Conditions include queries such as how full is the disk, how high is the patient's blood pressure, and how much revenue is the company generating today? Outcomes include such issues as is the server in trouble, are the users having a good time, is the situation safe? The challenge throughout the work was to develop a display technology that conveyed the importance of attending to a situation as well as to develop a reproducible method for capturing that importance.

A Three-Step Process: Identify, Describe, Refine

I propose a three-step process for any particular situation: identify, describe, refine. The refinement step is focused mostly on threshold editing. My testing demonstrates that this can be done and that it works well.

The description step is based on applying a modification of fuzzy set theory. Given that fuzzy set arithmetic is established and tested, and that the interface can convey the result of the arithmetic as well, I believe that this works well. My observations were limited, however, so this conclusion cannot be made with certainty.

The process of situation identification is the most difficult. It is well established in the literature on skilled performers that experts do not have conscious awareness of the actions they take. Years ago, Rasmussen (1986) labeled this “skill based behavior.” That phenomenon showed up throughout this project. What people monitor for is almost always something they've developed expertise in. This is a well-known problem, one that was continuously present in the development of Expert System several decades ago. In order to try to make their decision criteria explicit, I asked questions designed to elicit some information about their monitoring methodology. I asked what they were looking at and what they were looking for. If they could tell me what they were looking at, they would be able to tell about conditions that they were keeping an eye on. If they told me what they were looking for, they would be able to tell me about a situation they were concerned might arise. The former response meant that I would identify the situation in a bottom-up manner, accumulating other influencing or mitigating conditions until I could describe the situation. The latter response meant that I had to identify the situation in a top-down manner, clarifying what other situations or conditions would lead the subject to think that the situation had arisen. But the procedure was particularly difficult because subjects who monitored conditions were so focused on those conditions that they had a hard time extrapolating upward, and subjects who monitored broad situations were unable to describe the information they used.

I started this research with a classification model of monitoring problems based on two dimensions: the knowledge of the observer, and ability to sense all the data necessary to assess the situation. Hollnagel and Woods (2005) use pretty much the same model to describe where Joint Cognitive Systems are needed, but they add the dimension of time to make an assessment to their model. My original classification could still be applied to the problem of situation identification, where condition-focused subjects have both knowledge and the ability to directly sense those conditions, and situation-focused subjects have knowledge, but not the ability to directly sense. However, I don't think that classification would help someone trying to apply heed to their own problem. Focusing the classification on questioning how an observer monitors (what they do or are thinking about) helps others to adopt heed as a way of thinking about their own monitoring problems.

Throughout this work, the most difficult part was understanding what it is that subjects monitor and how it is that they actually do it. These studies make a start toward systematically identifying those situations, then describing each situation enough to capture it. The heed framework then provides tools to develop and maintain awareness.

Chapter 6: Discussion and Future Work

Introduction

In this dissertation I introduce the heed scale that serves as an indicator of the need for immediate attention to a situation. I introduce the set of transformations upon sensors and the fuzzy-logic-based algebra of combinations of sensor value to produce the heed scale. I develop an example interface that displays heed values, provides a historical trail and a predicted time line, where the predictions also display the relative likelihood of falling within the predicted range. I also provide a system for authoring the heed system's computations and decision logic. In this chapter I discuss the implications of heed for supporting situation awareness, where and how the method can be improved, a set of guidelines to aid interface designers, and how this method enables the development of computational artifacts.

Future Work

The interface and example applications are only prototypes; many improvements could make them more useful and more broadly applicable.

Improving the accuracy of heed forecast will be a valuable area for future work. One important area so far unexplored is that of scripts in the situation description hierarchy. Scripts are a time-ordered occurrence of schemas. The system, as implemented, employs only a very simple method of forecasting based on the recent history of sensor readings. The biggest drawback of this method is that it does not take into account long-term patterns. A forecast based on event pattern matching, or one of the other forecasting methods discussed in chapter 3 may lead to more accurate predictions.

The heed for a situation is a continuously changing value, and if observers do not have time to attend to a situation then a heed-based display will continue to remind them. But what if the heed diminishes before observers can attend to it? Because the observers may want to delve into that event to discover what had triggered the high heed value the interface must support a means of returning to the previous time. The interface, as implemented, does not support this kind of retrospective analysis. This is a clear area for study for future investigation.

The gizmometer interface is just one method of rendering heed values, and future work should explore alternative displays. As a simple scale, heed values lend themselves to a variety of mappings. Ambient displays where auditory signals are mapped to heed values are not hard to imagine, as are ambient displays that map heed values to physical attributes (perhaps a display that inflates and deflates balloons, or one that moves a set of physical dials).

How can heed be applied to a team environment where configurations are shared among observers? It's not difficult to imagine situations where similarly deployed systems will develop a community of experts who seek to share their acquired knowledge with one another. Alternatively, for a single deployment, over time, new employees will need to be trained on the system and it is likely that established expert observers of the system would want to share their understanding. In that case, how much does access to the expert influence the effectiveness of the interface? How well does an observer cope when the expert is unavailable, and how well does the expert cope with having to support the observer amidst other tasks? Future work should explore how heed can be applied in multi-operator environments.

Another application area that may be interesting to apply the concepts of heed to is application steering. Large-scale distributed applications, such as simulations, can benefit from adjusting their resource allocations from time to time. These large-scale applications run on shared computational resources that can be expensive to use. The resources that may be allocated for the application can depend on available budget, urgency to complete the application run, and the particular nature of the computation at various points in the application execution. Some applications can run for days at a time, and in those instances, it does not make sense for the user to monitor the application intently the entire time. Automating the process of resource allocation is made difficult because the use of resources can cost money, and making decisions in light of their financial impacts is still best left to users. A heed display could be used to identify situations that would indicate the application might need more resources soon, or to identify when an application might be wasting money on unnecessary resources.

Discussion

Based on the work described in this dissertation, I offer some guidelines for how monitoring tools should be designed, the activities they should support, and how heed can be used.

Situational vs. Conditional Monitoring

The observer can set up monitoring for specific conditions, but the studies with the four subjects demonstrate that people monitor for situations, not just conditions. They may focus on indicative conditions, but they eventually consider mitigating circumstances or contributing conditions before taking action. Subjects B and C both were interested in monitoring for situations, and not just individual conditions. The test case set up for Subject A was an example only focused on a few conditions, but even that subject could easily describe mitigating circumstances, such as during nightly backups when the server load would be expected to be high. To support the monitoring of situations, it is important to enable the user to see the status of contributing factors and mitigating conditions and to exclude information not directly relevant to the situation. Heed is an effective framework to implement displays that support situation monitoring, and future work could explore its effectiveness as a means to implement tools for situation awareness.

Eliciting Knowledge

In order to identify what is worth monitoring at any moment, the interface needs to know what situations the observer would want to attend to. How can it know what that is? Either the interface must incorporate an extensive knowledge base and act as an expert system, or the interface must enable observers to author that knowledge base themselves. Expert systems have been used in monitoring tools before, and there is a lot to be said for this approach. If a complex knowledge base can be built to encompass the complete performance parameters of a system, then an expert system would be a very powerful tool for predicting the condition of that system, which in turn would facilitate all manner of decision-making. However, there are two problems: building such knowledge base is time-consuming and very difficult, and while it's being built, the monitoring display is not useful.

In the previous chapter I discussed a top-down and a bottom-up design methodology to help build the knowledge base for a heed-based display. The advantage of this approach is that the application can be used while the knowledge base is being built. In fact, a central tenant of the heed framework is that the knowledge base for a monitoring application is never finished, because, in a complex system, the operating environment, the knowledge of the observer, or the system itself constantly changes. In order to underscore the notion that the information presented to the user is to be refined or adjusted, the information display itself doubles as a modification interface. Thus, even when the knowledge base is in early stages of development and therefore incomplete, it is treated as a working prototype by the observer to be refined over time. As a result, it can be used in support of monitoring tasks. The observer is encouraged to correct the knowledge base and thereby continually improve performance while enhancing trust in the values being displayed.

Evaluation vs. Analysis

Based on the work in this dissertation, I propose that monitoring interfaces distinguish between activities of evaluation and those of analysis. I define analysis as an activity that seeks to draw an objective conclusion and that is precise, conclusive, and reproducible. Evaluation, on the other hand, is imprecise, may be uncertain, and is unique to the individual and the situation: it seeks to draw a subjective conclusion. The heed scale and gizmometer display are ideally suited for evaluation.

Subject A was interested in knowing when his servers were heavily loaded — not when they were exactly 92% loaded. However, he learned what the typical load levels were after analyzing the configuration information recorded by the interface. Subject C was interested in knowing if and when his users were having a negative experience — something that is inherently subjective. By also analyzing the information stored by the interface as well as data recorded by his servers he learned more about his users. In these two examples, the activities that lead to an evaluation and those that lead to analysis are distinct. The use of imprecise terminology is not the hallmark of evaluation, but it can be helpful in distinguishing if an evaluative or analytical tool is needed. A user-centered design approach will help determine when an observer is interested in evaluative information and when analytic information is called for. Noting that people naturally use imprecise descriptions to describe conditions was, in part, the impetus for fuzzy set theory (Zadeh 1978), and is also why sensor conditions employ a similar definition.

My subjects also highlight a problem with current tools: they are designed to support analysis, but are used for evaluation as well. Subject A had used the UNIX uptime command when all he really wanted to know was if the servers were loaded. Subject C had access to detailed website monitoring tools intended to monitor page access and bandwidth usage, but those tools were unsatisfactory to him because he found himself becoming too involved with the numbers instead of what they meant. Because the tools were designed for analysis, he resorted to analysis, even when he just wanted a simple evaluation of the situation.

Each type of activity has different requirements. Analysis activities require detailed and precise information and are performed carefully, while evaluation can tolerate imprecision and is performed as quickly as possible. System observers need tools for evaluation to supplement analytical tools. Alarms and dashboards of graphs, dials, or colored indicators are well-suited to analytical tasks, such as investigating problems or optimizing performance, but the detail they provide can inhibit making evaluations. Complex systems require interfaces for both.

A heed-based display is intended as a supplementary system to an existing alarm system or to an existing report generating or sensor dashboard system. It supplements an existing alarm system by providing an interface that indicates to an observer how soon an alarm will be triggered. By configuring the display's sensor condition with the attend boundary set to the same value as an existing alarm threshold, and the ignore boundary set to a nominal reading, then the display can be used to indicate if an alarm might be triggered soon. A heed-based display can further support an alarm system by indicating the heed of situations that are known to, or thought to, contribute to an alarm trigger. However, without an alarm system, a heed-based display does not actively disrupt an observer's activity and force him or her to glance at the display. Although there is no reason that a heed-based display could not include an alarm itself, the problem of alarm proliferation has already been discussed in chapter 2, and a heed-based display would do best not to contribute to that problem further.

A heed-based display also supplements a report-generating system or a sensor dashboard. These two systems provide observers with significant detail and facilitate the analysis of situations and their causes. However, both are time-consuming to read and parse. Dashboards and reports combine information of different forms and visualizations and are presented in their native units of measurement. This high level of detail requires the observer to interpret and evaluate each aspect of the report or dashboard that they deem as relevant to the situation they are monitoring. As a result, the observer must process and remember many things just to assess the situation: which aspects to focus and which to ignore, what each sensor read previously, when the last time was that they looked into the situation, and much more.

Reports can synthesize a great deal of data, making them more concise than raw sensor data dashboards, but a report must be read to determine the condition of the situation it presents, even if it is nominal. Reports can take a great deal of time to read, and frequently, such reading only reveals that the situation is unchanged or nominal. As a result, reports are often ignored or set aside. A heed-based display takes much less time to read and, like a report, can synthesize many sensor readings.

Reports and dashboards are excellent for analysis and scrutiny of data, but because they are time-consuming to read and parse, they cannot be used to maintain awareness without significant effort and vigilance. A heed-based display requires less effort to read and interpret than a dashboard or report because it only conveys one dimension and can synthesize many sensor readings into a single display. However, a heed-based display should not be used without a dashboard, a report, or some other raw sensor display. A situation with high heed requires attention, and that attention must first be paid by investigating or confirming the situation. Furthermore, a heed-based display also benefits from the presence of an alarm system. Even though a heed-based display can convey more information than an alarm can, all three systems, in conjunction, enable an observer to maintain awareness of situations, react to events, and analyze each occurrence.

Monitoring: What To Ignore And What To Heed

As Norman (2007) points out, monitoring is as much about what to ignore as it is about what to pay attention to. People who monitor complex systems, like the four subjects studied, all have many other responsibilities. Monitoring tools should enable users to confidently continue to work on what they are already doing, and not just tell them what to turn their attention to. Monitoring interfaces must not only identify the situations that observers should attend to, but also simplify confirming that a situation does not need attention. This, in part, is how Subject B used the interface; by confirming to her that she did not need to read her reports yet. Furthermore, when possible, a monitoring interface should help them decide how much longer an observer can afford to ignore a situation. A forecast as part of the display can help in that regard.

Heed-Based Display As A Cognitive Artifact

By including authorship in their design, monitoring tools can be used as a cognitive artifact enabling observers to follow up on hunches, or to help them witness unusual events, not just explore those events in hindsight when it is no longer possible to assert control over the system. In order to do that, the tools must make it simple to define a suspicious set of circumstances. Then the tools can be used as an external memory for those circumstances as well as helping to "keep an eye out" for the event to occur or resurface.

A monitoring system is not just about identifying problems before they arrive, but identifying opportunities before you miss them as well. An observer can also define rare occurrences directly using heed to give warning of an impending incident so that there may be time to observe the event. An important aspect of learning about a complex system is experimenting with it. Although directly experimenting with a live system is rarely a good idea, having the ability to define a set of circumstances that would indicate suspicious or rare conditions can enable the observer to identify an opportunity when it's best to observe the system. When a bottleneck had eluded him during pre-deployment testing, this is what subject A wanted to be able to do: watch his servers while they were loaded to find that bottleneck, but not once the problem had been found, analyzed, and corrected. Instead he and his whole staff had to spend the day engrossed in monitoring the servers.

Conclusion

These studies demonstrate the role of heed, a simple graphical display (the gizmometer), and an authoring tool in support of the monitoring of complex systems. By designing a monitoring system to help people know how to allocate their attention, rather than around alarms that require mitigating action, we can build interfaces that enable monitoring while respecting the limited time and resources of human observers. This design goal also changes the nature of the communication from the system to the observer, conveying what the system has identified as worth attending to instead of to what requires action ("Hey, take a look at this" instead of "Hey, take care of this."). Moreover, the authoring tool permits the observer to communicate with the system. Together, this two-way communication between observer and system enables a creative, effective human-computer relationship.

References

Arrabito, G. R., T. A. Mondor, et al. (2004). "Judging the urgency of non-verbal auditory alarms: a case study." Ergonomics **47**(8): 821 - 840.

Auxer, G. A., J. F. Carroll, et al. (2007). Rating that represents the status along a specified driving route. USPTO. US, Traffic.com, Inc.

Banbury, S. P., W. J. Macken, et al. (2001). "Auditory Distraction and Short-Term Memory: Phenomena." Human Factors **43**(1): 12.

Belz, S. M., G. S. Robinson, et al. (1999). "A New Class of Auditory Warning Signals for Complex Systems: Auditory Icons." Human Factors **41**(4): 608.

Belz, S. M., J. J. Winters, et al. (1997). "A Methodology for Selecting Auditory Icons for Use in Commercial Motor Vehicles." Human Factors and Ergonomics Society Annual Meeting Proceedings **41**: 939-943.

Bianchi, M., M. Boyle, et al. (1999). "A comparison of methods for trend estimation." Applied Economics Letters **6**(2): 103 - 109.

Bliss, J. P. and R. D. Gilson (1998). "Emergency signal failure: implications and recommendations." Ergonomics **41**: 57-72.

Bliss, J. P., R. D. Gilson, et al. (1995). "Human probability matching behaviour in response to alarms of varying reliability." Ergonomics **38**(11): 2300 - 2312.

Brezinski, C. and M. R. Zaglia (1991). Extrapolation Methods. Theory and Practice. Amsterdam, North-Holland Publishing Co.

Breznitz, S. (1984). Cry wolf : the psychology of false alarms. Hillsdale, N.J., Lawrence Erlbaum Associates.

Buharali, A. M. and T. B. Sheridan (1982). Fuzzy set aids for telling a computer how to decide, United States, IEEE,New York, NY.

Butler, K. A., J. Zhang, et al. (2007). Work-centered design: a case study of a mixed-initiative scheduler. Proceedings of the SIGCHI conference on Human factors in computing systems. San Jose, California, USA, ACM.

Cutting, J. E. (2002). "Representing motion in a static image: constraints and parallels in art, science, and popular culture." Perception **31**: 1165-1193.

Deming, W. E. (1975). "On Probability as a Basis for Action." The American Statistician **29**(4): 146-152.

Draper, N. R. and H. Smith (1998). Applied Regression Analysis. New York, Wiley.

Edworthy, J. (1994). Urgency Mapping in Auditory Warning Signals. Human Factors in Alarm Design. N. Stanton. Bristol, PA, Taylor & Francis**:** 15-30.

Endsley, M. R. (1988). Design and Evaluation for Situation Awareness Enhancement. Proceedings of the Human Factors Society 32nd Annual Meeting, Santa Monica, CA.

Endsley, M. R. (1995). "Toward a Theory of Situation Awareness in Dynamic Systems." Human Factors: The Journal of the Human Factors and Ergonomics Society **37**: 32-64.

Endsley, M. R., B. Bolté, et al. (2003). Designing for Situation Awareness: An Approach to User-Centered Design, Taylor & Francis.

Feuer, A. and G. C. Goodwin (1996). Sampling in digital signal processing and control, Birkhauser Boston Inc.

Frohlich, D. M. (1993). "The history and future of direct manipulation." Behaviour & Information Technology **12**(6): 315 - 329.

Fu, W.-T. and J. Anderson (2008). "Solving the credit assignment problem: explicit and implicit learning of action sequences with probabilistic outcomes." Psychological Research **72**(3): 321-330.

Hollnagel, E. and D. D. Woods (2005). Joint Cognitive Systems: Foundations of Cognitive Systems Engineering. Boca Raton, FL, CRC Press.

Hutchins, E. L., J. D. Hollan, et al. (1985). "Direct Manipulation Interfaces." Human-Computer Interaction **1**(4): 311 - 338.

Kahneman, D. and A. Tversky (1979). "Prospect Theory: An Analysis of Decision under Risk." Econometrica **47**(2): 263-291.

Lehto, M. R., Papastavrou, J. D., & Giffen, W. J. (1998). "An empirical study of adaptive warnings: Human- versus computer-adjusted warning thresholds." International Journal of Cognitive Ergonomics(2): 19-33.

Leung, Y. K. and M. D. Apperley (1994). "A review and taxonomy of distortion-oriented presentation techniques." ACM Trans. Comput.-Hum. Interact. **1**(2): 126-160.

McCloud, S. (1993). Understanding comics. Northampton, MA, Kitchen Sink Press.

Mobley, R. K. (2002). An Introduction to Predictive Maintenance, Butterworth-Heinemann.

Montgomery, D. C. (2008). Introduction to statistical quality control. Hoboken, N.J., Wiley ; Chichester : John Wiley.

Nelson, L. S. (1984). "Technical Aids." Journal of Quality Technology **16**(4): 238-239.

Ni, J., J. Lee, et al. (2003). "Watchdog Agent--an infotronics-based prognostics approach for product performance degradation assessment and prediction." Advanced Engineering Informatics **17**(3-4): 109-125.

Norman, D. A. (1990). "The 'Problem' with Automation: Inappropriate Feedback and Interaction, Not 'Over-Automation'." Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences **327**(1241): 585-593.

Norman, D. A. (2007). The design of future things, Basic Books.

Nowlan, F. S. and H. F. Heap (1978). Reliability-Centered Maintenance, UNITED AIR LINES INC SAN FRANCISCO CA**:** 515.

Parasuraman, R. and V. Riley (1997). "Humans and automation: use, misuse, disuse, abuse." Human Factors **v39**(n2): p230(24).

Rasmussen, J. (1986). Information Processing and Human-Machine Interaction. New York, Elsevier Science Publishing Co.

Ren, J. and T. B. Sheridan (1994). Optimization with fuzzy linear programming and fuzzy knowledge base. Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the Third IEEE Conference on.

Ren, J. and T. B. Sheridan (1995). "An adaptive decision aid for real environments." Systems, Man and Cybernetics, IEEE Transactions on **25**(10): 1384-1391.

Rundle, J. B., K. F. Tiampo, et al. (2002). "Self-Organization in Leaky Threshold Systems: The Influence of Near-Mean Field Dynamics and Its Implications for Earthquakes, Neurobiology, and Forecasting." Proceedings of the National Academy of Sciences of the United States of America **99**(3): 2514-2521.

Sheridan, T. B. (1976). Monitoring Behavior and Supervisory Control, Perseus Publishing.

Shewart, W. A. (1931). Economic control of quality of manufactured product. New York, Van Nostrand.

Simon, D. (2006). Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches, Wiley-Interscience.

Sorkin, R. D., B. H. Kantowitz, et al. (1988). "Likelihood alarm displays." Human Factors **30**(4): 445-459.

Sorkin, R. D. and D. D. Woods (1985). "Systems with Human Monitors: A Signal Detection Analysis." Human-Computer Interaction **1**(1): 49 - 75.

Stanton, N. (1994). Alarm initiated activities. Human factors in alarm design, Taylor & Francis, Inc.**:** 93-117.

Stanton, N. A. and R. B. Stammers (1998). "Alarm Initiated Activities: matching formats to tasks." International Journal of Cognitive Ergonomics **2**(4): 331-348.

Sterman, J. D. (1994). "Learning in and about complex systems." System Dynamics Review **10**(2-3): 291-330.

Thomas, B. S. (1992). Telerobotics, automation, and human supervisory control, MIT Press.

Wall, L. (2000). Programming Perl, O'Reilly &Associates, Inc.

Western Electric, C. (1956). Statistical quality control handbook. Indianapolis, Western Electric Co.

Wilkinson, R. T. (1964). "Artificial 'Signals' As An Aid To An inspection Task." Ergonomics **7**(1): 63 - 72.

Wogalter, M. S., D. M. Dejoy, et al. (1999). Organizing Theoretical Framework: A Consolidated Communication-Human Information Processing (C-HIP) Model. Warnings and Risk Communication. M. S. Wogalter, D. M. Dejoy and K. R. Laughery. Philadelphia, PA, Taylor & Francis**:** 15-23.

Young, S. L. and D. R. Lovvoll (1999). Intermediate Processing Stages: Methodological Considerations for Research on Warnings. Warnings and Risk Communication. M. S. Wogalter, D. M. Dejoy and K. R. Laughery. Philadelphia, PA, Taylor & Francis**:** 27-52.

Zadeh, L. A. (1965). "Fuzzy sets [J]." Information and Control **8**(3): 338-353.

Zadeh, L. A. (1978). "Fuzzy sets as a basis for a theory of possibility." Fuzzy Sets and Systems **1**(1): 3-28.

Zimmermann, H. J. (2001). Fuzzy Set Theory--and Its Applications. Boston, Kluwer Academic Publishers.

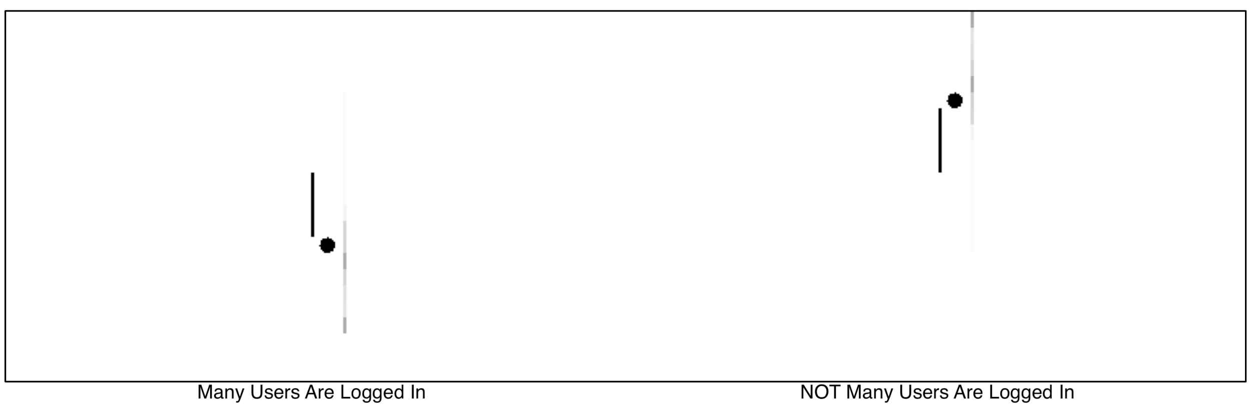
Zimmermann, H. J. and P. Zysno (1980). "Latent connectives in human decision making." Fuzzy Sets and Systems **4**(1): 37-51.

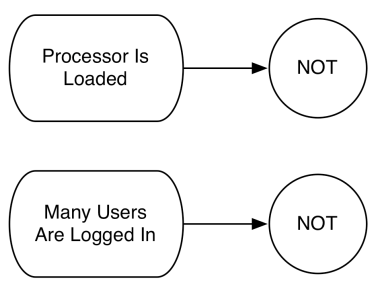
Appendix A

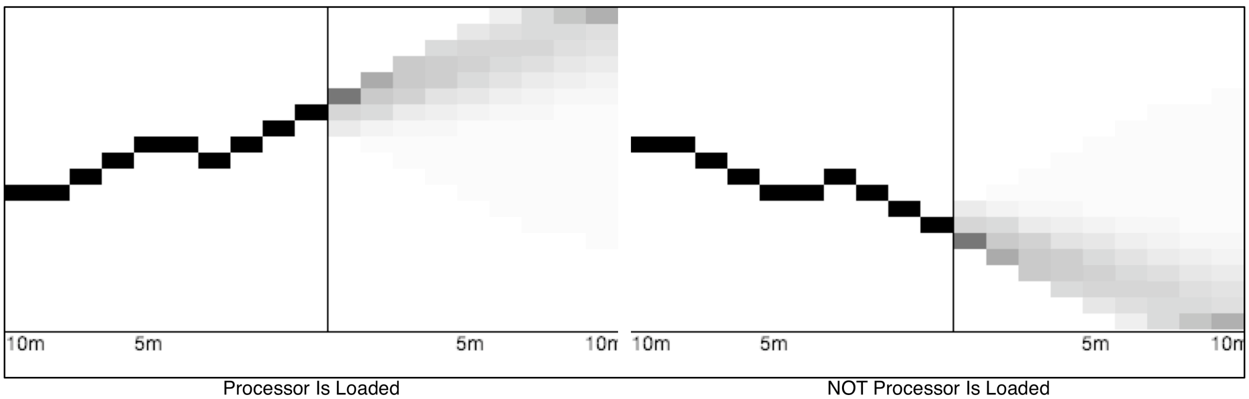
Examples of Heed Arithmetic

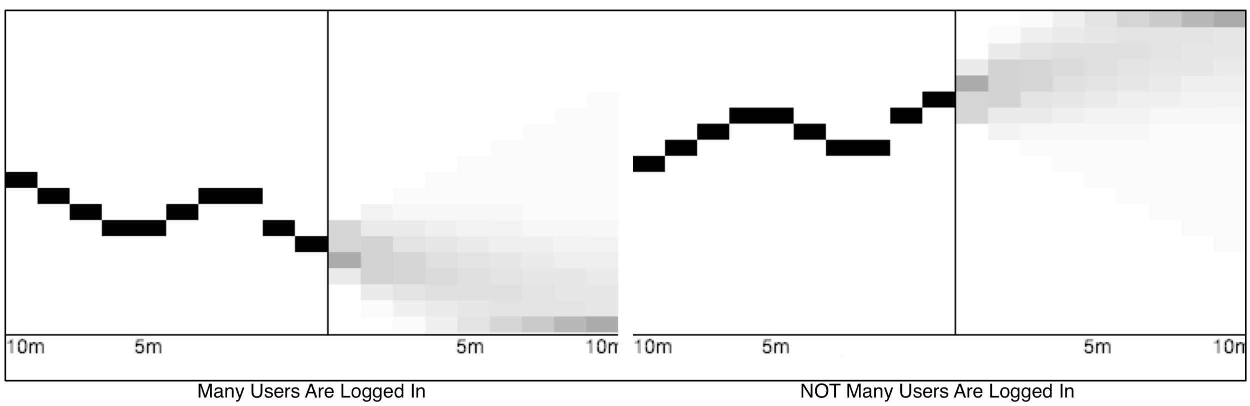
This appendix illustrates, by way of example, the logical operators described in chapter 3. First the example conditions are shown, each with their NOT applied inverse. Next, the sensor condition's OR combination is shown, then their AND combination, and finally, their AND-N combination. The AND-N combination is shown twice to highlight that the AND-N operator is non-associative. Each slider view is followed by its corresponding graph view.

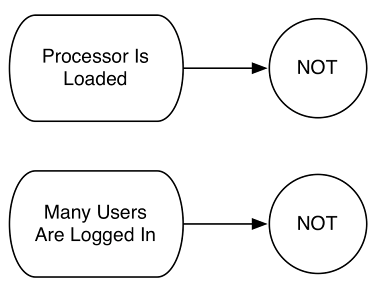

Processor Is Loaded
NOT Processor Is Loaded


Many Users Are Logged In
NOT Many Users Are Logged In

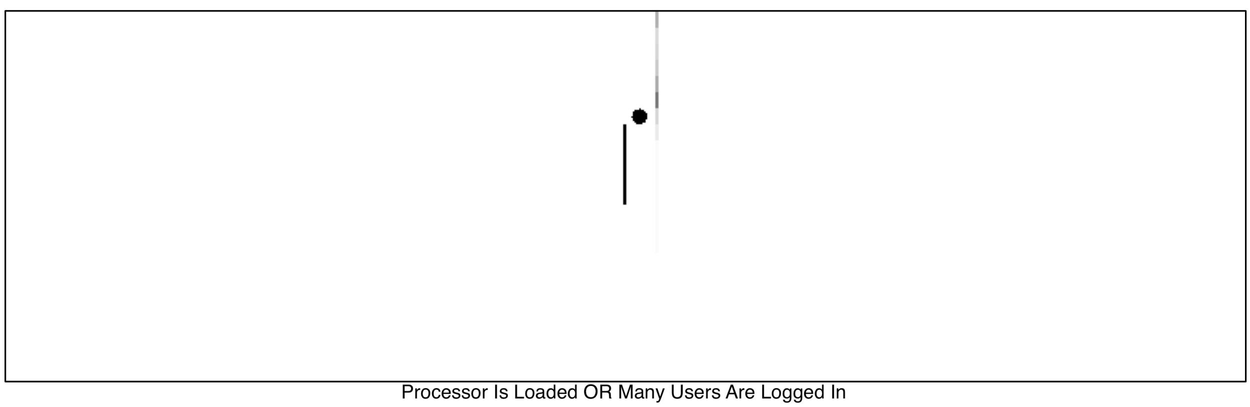

Processor Is Loaded
NOT
Many Users Are Logged In
NOT

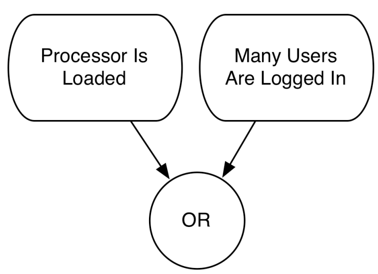

10m
5m
5m
10m
10m
5m
5m
10m
Processor Is Loaded
NOT Processor Is Loaded

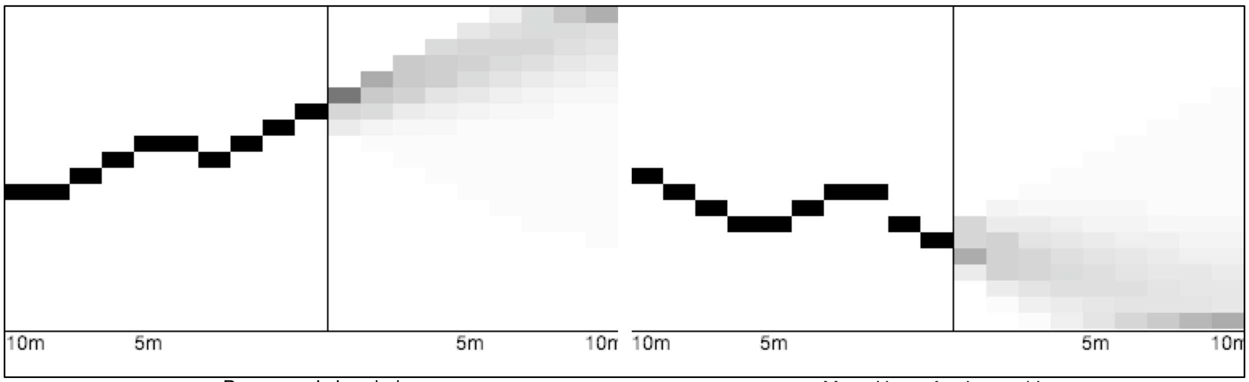

10m
5m
5m
10m
10m
5m
5m
10m
Many Users Are Logged In
NOT Many Users Are Logged In


Processor Is Loaded
NOT
Many Users Are Logged In
NOT


Processor Is Loaded
Many Users Are Logged In

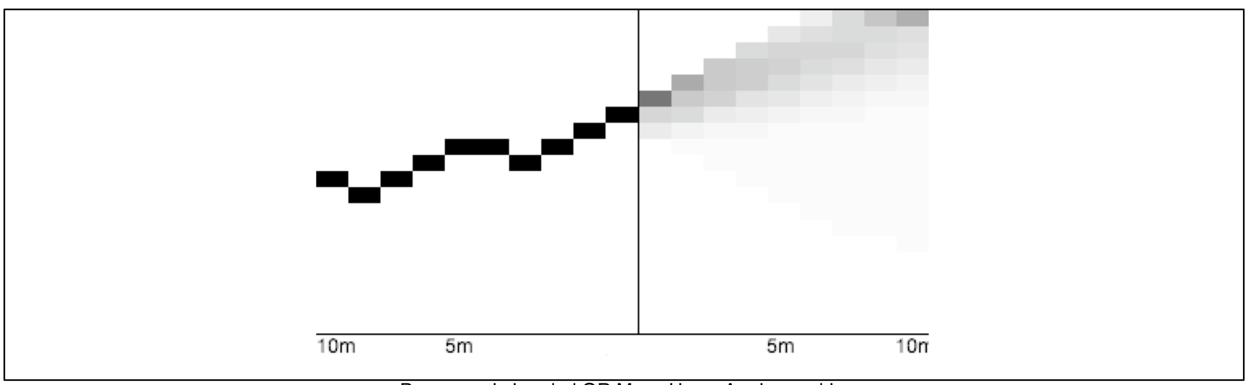

Processor Is Loaded OR Many Users Are Logged In


Processor Is Loaded
Many Users Are Logged In
OR

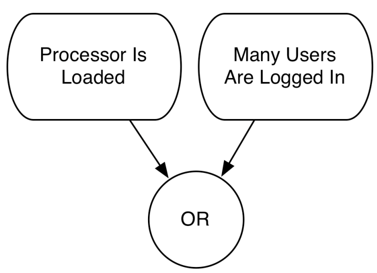


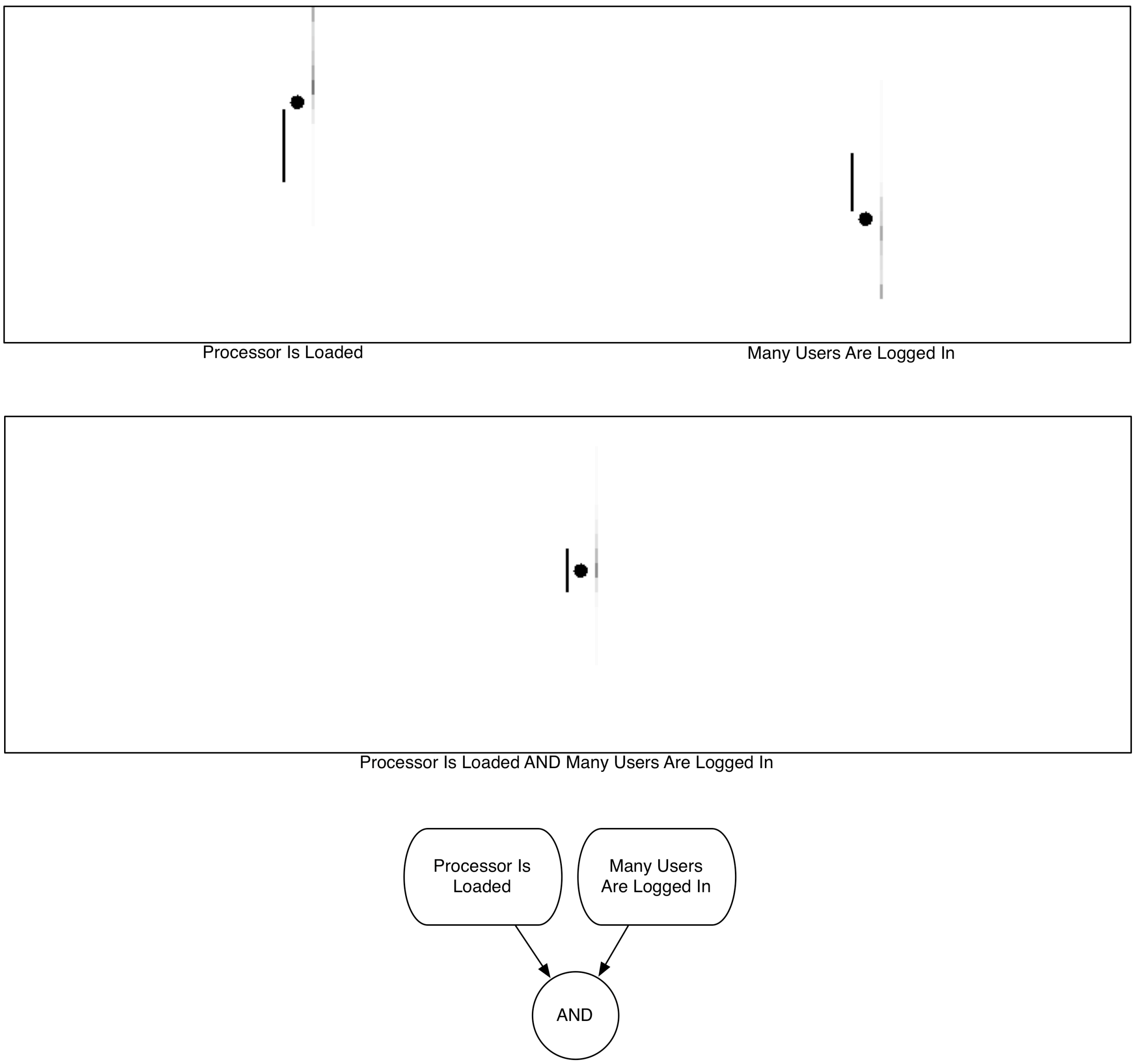
Processor Is Loaded

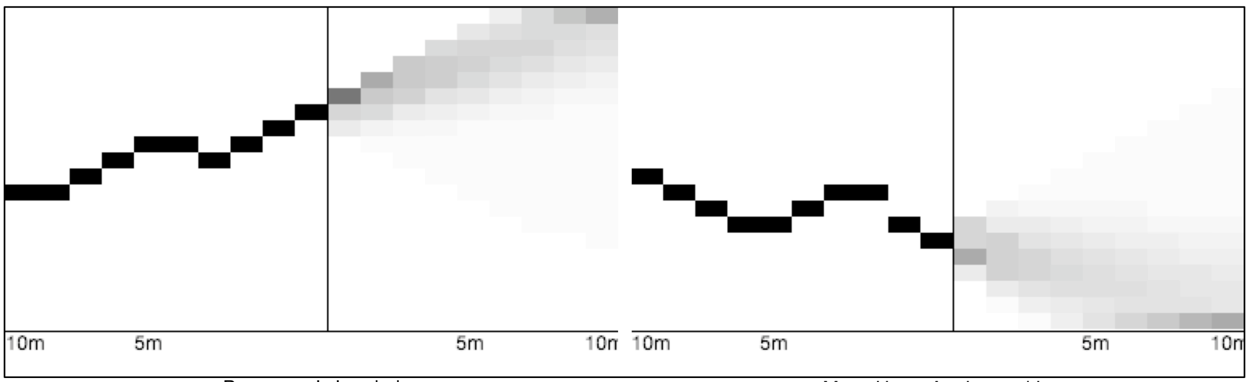
Many Users Are Logged In



Processor Is Loaded OR Many Users Are Logged In

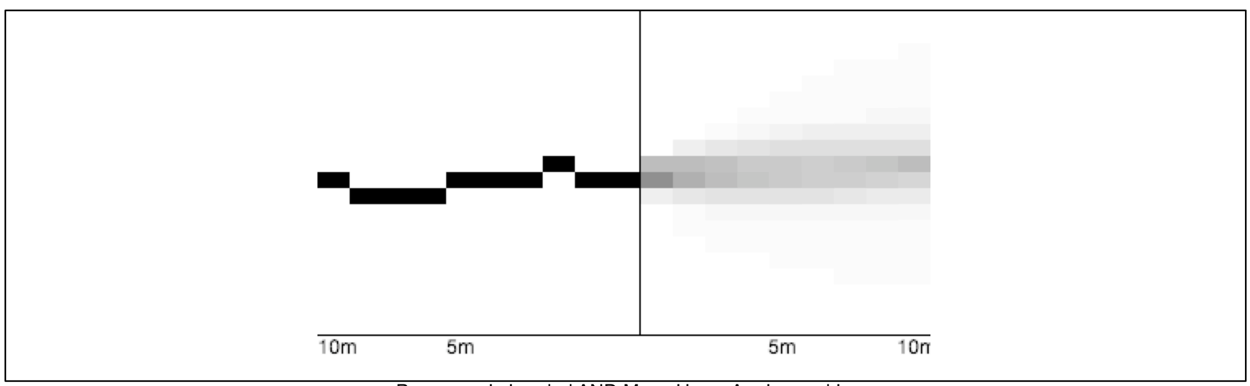



Processor Is Loaded
Many Users Are Logged In
Processor Is Loaded AND Many Users Are Logged In
Processor Is Loaded
Many Users Are Logged In
AND

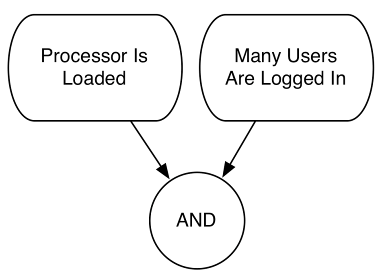


Processor Is Loaded

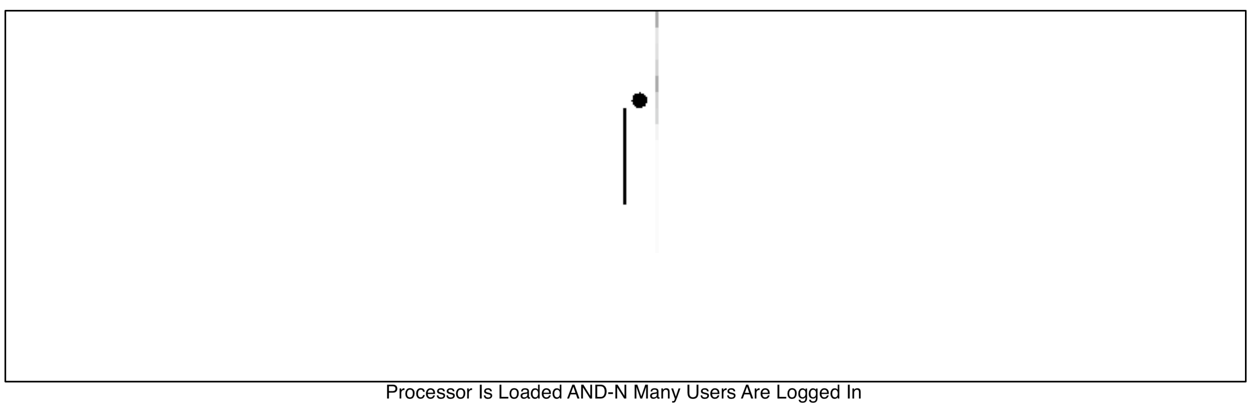
Many Users Are Logged In

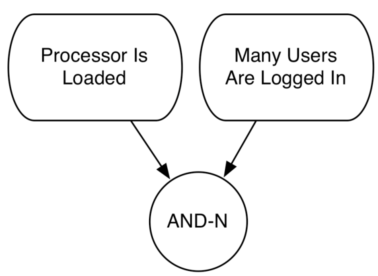


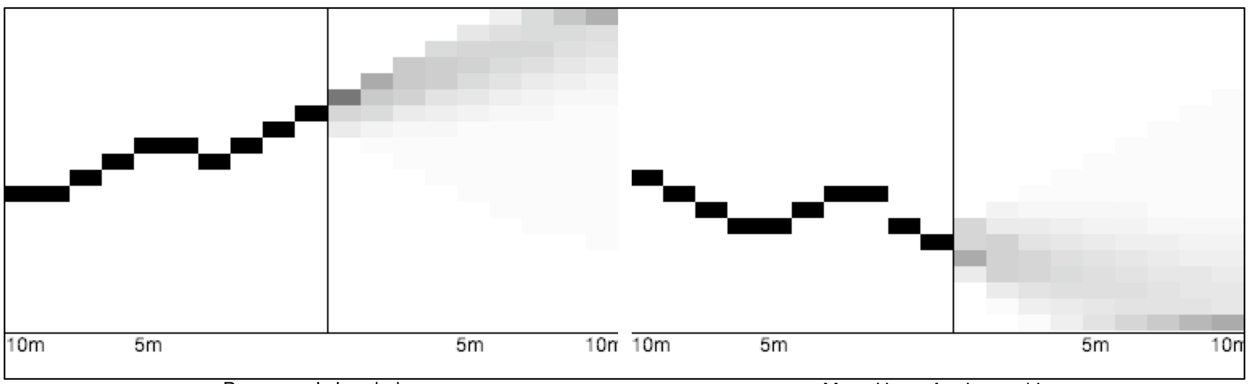
Processor Is Loaded AND Many Users Are Logged In



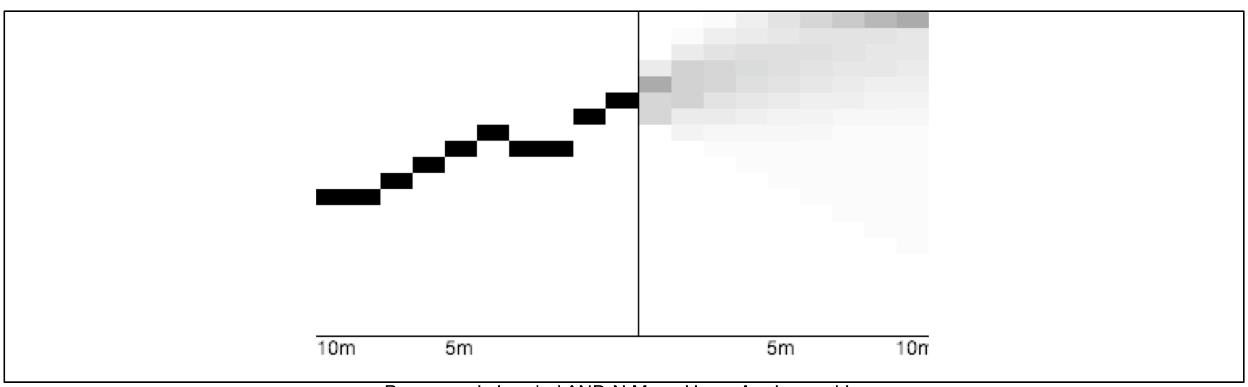

Processor Is Loaded
Many Users Are Logged In


Processor Is Loaded AND-N Many Users Are Logged In

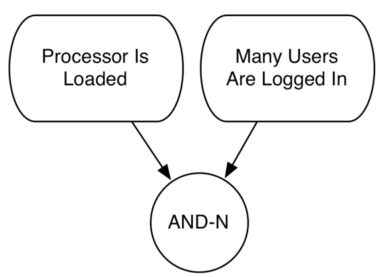

Processor Is Loaded
Many Users Are Logged In
AND-N



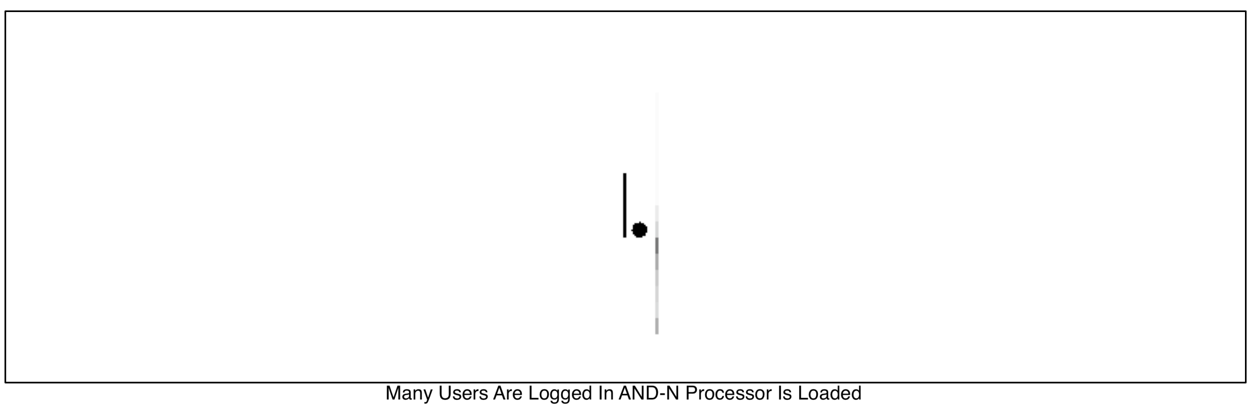
Processor Is Loaded Many Users Are Logged In

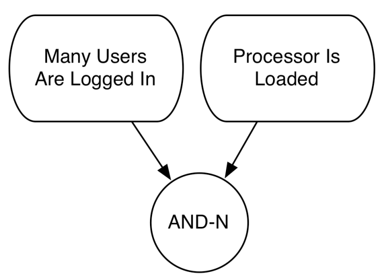


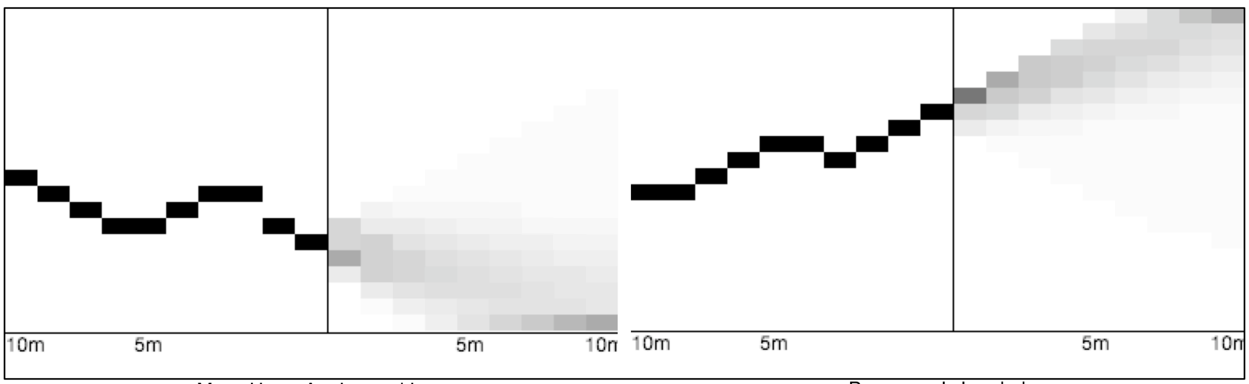
Processor Is Loaded AND-N Many Users Are Logged In



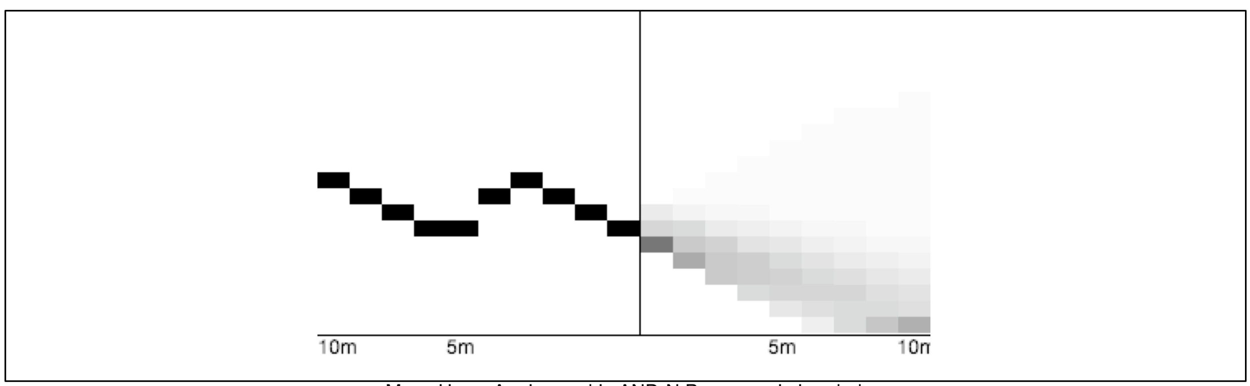

Many Users Are Logged In
Processor Is Loaded


Many Users Are Logged In AND-N Processor Is Loaded

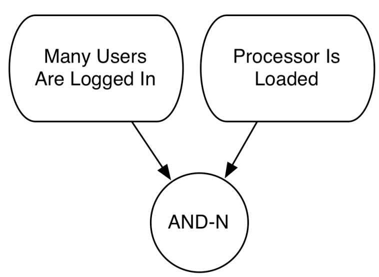

Many Users Are Logged In
Processor Is Loaded
AND-N



Many Users Are Logged In　　Processor Is Loaded



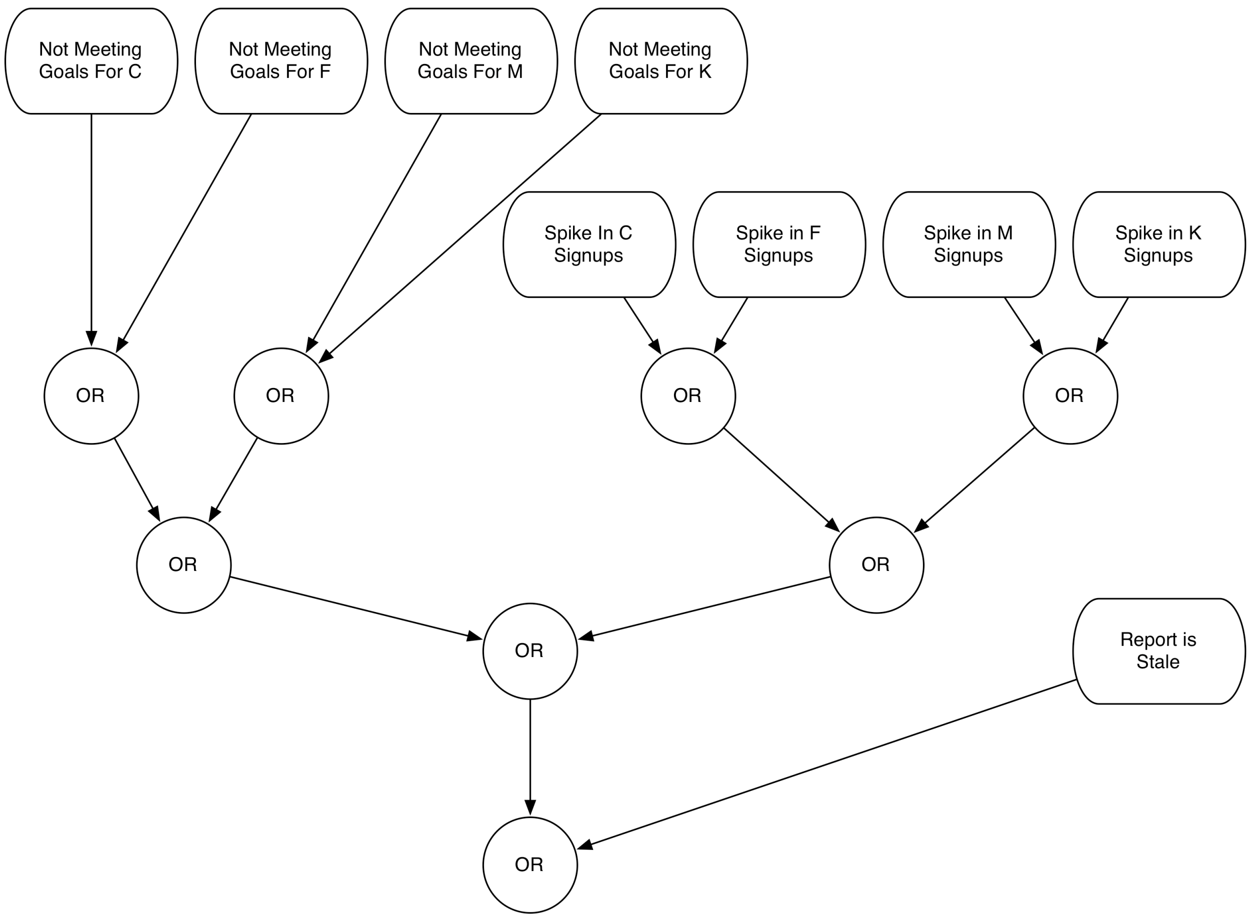
Many Users Are Logged In AND-N Processor Is Loaded



Appendix B

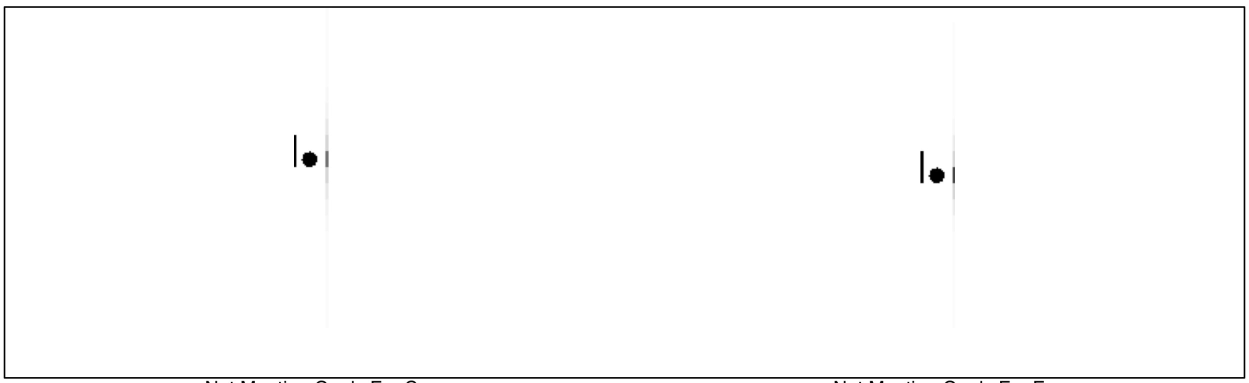
A Visual Example of a Heed Schema

This appendix elaborates an example heed display, namely the one shown in figure 29, by illustrating how a drill-down through a schema would appear. The schema, which is a simple OR-composite of nine sensor conditions, is shown first as a binary tree as well as its corresponding XML configuration file. Then the figures traverse the binary tree associated with each node, and ends with the root node. Each OR-node is shown with its antecedent leaves, and each slider view is followed by its corresponding graph view. The data, configuration file, and visuals are taken from the experiment conducted with subject B, but with identifying information made anonymous. The situation described in the schema for subject B identifies when a particular report related to a corporate program should be read. The report covers data related to four sectors associated with the corporate program. The subject should read the report when it has not been read in a while (i.e. when the last report read becomes stale), or when the program is not meeting the subject's financial goals in any of the four sectors, or when there is a spike in signups in any of the four sectors.



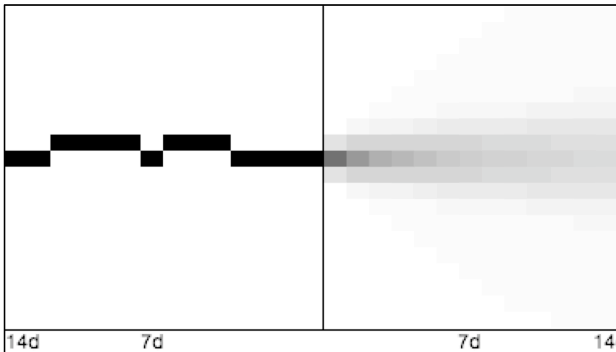
Not Meeting Goals For C
Not Meeting Goals For F
Not Meeting Goals For M
Not Meeting Goals For K
Spike In C Signups
Spike in F Signups
Spike in M Signups
Spike in K Signups
OR
OR
OR
OR
OR
OR
OR
Report is Stale
OR

```xml
<?xml version="1.0" ?>
<gizmometer-config>
    <sensor-databases>
        <database DESCRIPTION="Local"
            DATABASE="sensor_db" LOCATION="127.0.0.1" TYPE="MySQL"
            USER="heed_user" PASSWORD="the_db_pass"/>
    </sensor-databases>
    <sensor-definitions>
        <sensor
            DESCRIPTION="C Daily Change" DATABASE="Local"
            POLLING-FREQ="86400" READING-MAX="4000" READING-MIN="0"/>
        <sensor
            DESCRIPTION="F Daily Change" DATABASE="Local"
            POLLING-FREQ="86400" READING-MAX="3500" READING-MIN="0"/>
        <sensor
            DESCRIPTION="M Daily Change" DATABASE="Local"
            POLLING-FREQ="86400" READING-MAX="2000" READING-MIN="0"/>
        <sensor
            DESCRIPTION="K Daily Change" DATABASE="Local"
            POLLING-FREQ="86400" READING-MAX="50000" READING-MIN="0"/>
        <sensor
            DESCRIPTION="C Weekly Growth Rate Deviation" DATABASE="Local"
            POLLING-FREQ="86400" READING-MAX="4000" READING-MIN="0"/>
        <sensor
            DESCRIPTION="F Weekly Growth Rate Deviation" DATABASE="Local"
            POLLING-FREQ="86400" READING-MAX="3500" READING-MIN="0"/>
        <sensor
            DESCRIPTION="M Weekly Growth Rate Deviation" DATABASE="Local"
            POLLING-FREQ="86400" READING-MAX="42000" READING-MIN="0"/>
        <sensor
            DESCRIPTION="K Weekly Growth Rate Deviation" DATABASE="Local"
            POLLING-FREQ="86400" READING-MAX="50000" READING-MIN="0"/>
        <sensor
            DESCRIPTION="Stale" DATABASE="Local"
            POLLING-FREQ="86400" READING-MAX="100" READING-MIN="0"/>
    </sensor-definitions>
    <condition-transformations>
        <transformation DESCRIPTION="linear"
          CLASS="LinearConditionTransformation"/>
    </condition-transformations>
    <sensor-conditions>
        <condition DESCRIPTION="Spike in C Signups"
            SENSOR="C Daily Change" TRANSFORMATION="linear"
            READING-MAX-HEED="5"
            READING-MIN-HEED="0"
            HISTORY-SAMPLES="14" FORECAST-SAMPLES="14" RANGE-SAMPLES="20"/>
        <condition DESCRIPTION="Spike in F Signups"
            SENSOR="F Daily Change" TRANSFORMATION="linear"
            READING-MAX-HEED="5"
            READING-MIN-HEED="0"
            HISTORY-SAMPLES="14" FORECAST-SAMPLES="14" RANGE-SAMPLES="20"/>
        <condition DESCRIPTION="Spike in M Signups"
            SENSOR="M Daily Change" TRANSFORMATION="linear"
            READING-MAX-HEED="5"
            READING-MIN-HEED="0"
            HISTORY-SAMPLES="14" FORECAST-SAMPLES="14" RANGE-SAMPLES="20"/>
        <condition DESCRIPTION="Spike in K Signups"
            SENSOR="K Daily Change" TRANSFORMATION="linear"
            READING-MAX-HEED="5"
            READING-MIN-HEED="0"
            HISTORY-SAMPLES="14" FORECAST-SAMPLES="14" RANGE-SAMPLES="20"/>
        <condition DESCRIPTION="Not Meeting Goals for C"
            SENSOR="C Weekly Growth Rate Deviation" TRANSFORMATION="linear"
            READING-MAX-HEED="0"
            READING-MIN-HEED="80"
            HISTORY-SAMPLES="14" FORECAST-SAMPLES="14" RANGE-SAMPLES="20"/>
        <condition DESCRIPTION="Not Meeting Goals for F"
            SENSOR="F Weekly Growth Rate Deviation" TRANSFORMATION="linear"
            READING-MAX-HEED="0"
            READING-MIN-HEED="80"
            HISTORY-SAMPLES="14" FORECAST-SAMPLES="14" RANGE-SAMPLES="20"/>
        <condition DESCRIPTION="Not Meeting Goals for M"
            SENSOR="M Weekly Growth Rate Deviation" TRANSFORMATION="linear"
            READING-MAX-HEED="0"
            READING-MIN-HEED="50"
            HISTORY-SAMPLES="14" FORECAST-SAMPLES="14" RANGE-SAMPLES="20"/>
        <condition DESCRIPTION="Not Meeting Goals for K"
            SENSOR="K Weekly Growth Rate Deviation" TRANSFORMATION="linear"
            READING-MAX-HEED="800.0"
            READING-MIN-HEED="507.11111111111074"
            HISTORY-SAMPLES="14" FORECAST-SAMPLES="14" RANGE-SAMPLES="20"/>
        <condition DESCRIPTION="Report Stale"
            SENSOR="Stale" TRANSFORMATION="linear"
            READING-MAX-HEED="5.0"
            READING-MIN-HEED="0.0"
            HISTORY-SAMPLES="14" FORECAST-SAMPLES="14" RANGE-SAMPLES="20"/>
    </sensor-conditions>
    <sensor-schema>
        <schema DESCRIPTION="Read Corporate Program Report">
            <or>
                <schema-condition CONDITION="Report Stale"
                    CRITICALITY="1.0" URGENCY="1.0"/>
                <paren>
                    <or>
                        <paren>
                            <or>
                                <paren>
                                    <or>
                                    <schema-condition
                                    CONDITION="Spike in C Signups"
                                    CRITICALITY="1.0" URGENCY="1.0"/>
                                    <schema-condition
                                    CONDITION="Spike in F Signups"
                                    CRITICALITY="1.0" URGENCY="1.0"/>
                                    </or>
                                </paren>
                                <paren>
                                    <or>
                                    <schema-condition
                                    CONDITION="Spike in M Signups"
                                    CRITICALITY="1.0" URGENCY="1.0"/>
                                    <schema-condition
                                    CONDITION="Spike in K Signups"
                                    CRITICALITY="1.0" URGENCY="1.0"/>
                                    </or>
                                </paren>
                            </or>
                        </paren>
                        <paren>
                            <or>
                                <paren>
                                    <or>
                                    <schema-condition
                                    CONDITION="Not Meeting Goals for C"
                                    CRITICALITY="1.0" URGENCY="1.0"/>
                                    <schema-condition
                                    CONDITION="Not Meeting Goals for F"
                                    CRITICALITY="1.0" URGENCY="1.0"/>
                                    </or>
                                </paren>
                                <paren>
                                    <or>
                                    <schema-condition
                                    CONDITION="Not Meeting Goals for M"
                                    CRITICALITY="1.0" URGENCY="1.0"/>
                                    <schema-condition
                                    CONDITION="Not Meeting Goals for K"
                                    CRITICALITY="1.0" URGENCY="1.0"/>
                                    </or>
                                </paren>
                            </or>
                        </paren>
                    </or>
                </paren>
            </or>
        </schema>
    </sensor-schema>
    <sensor-script/>
    <display>
        <slider SCHEMA="Read Corporate Program Report" UPDATE-FREQ="86400"/>
    </display>
</gizmometer-config>
```

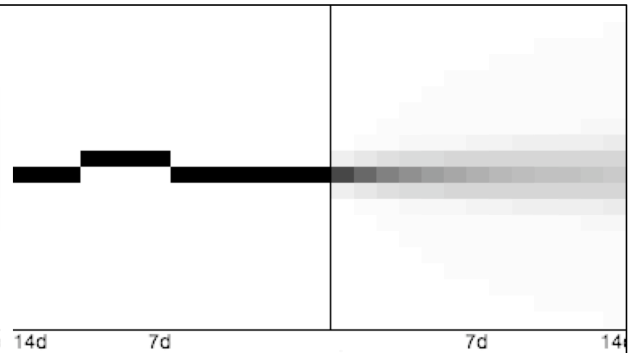


Not Meeting Goals For C Not Meeting Goals For F

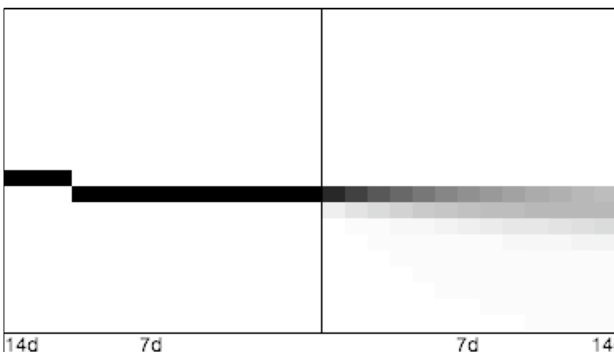
Not Meeting Goals For M Not Meeting Goals For K



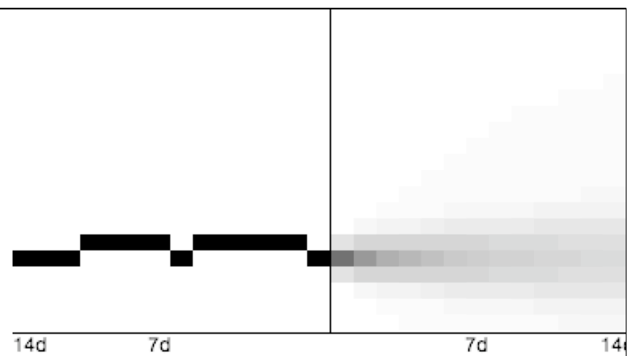
Not Meeting Goals For C



Not Meeting Goals For F



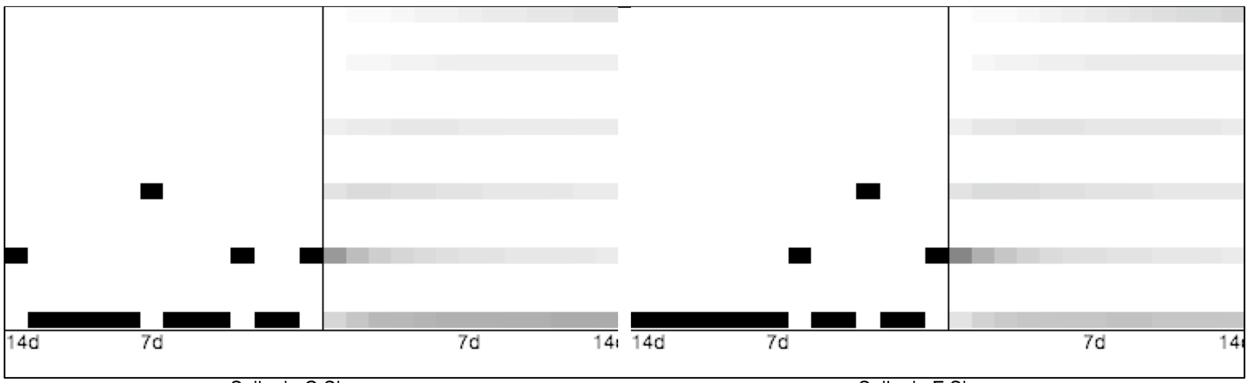
Not Meeting Goals For M



Not Meeting Goals For K

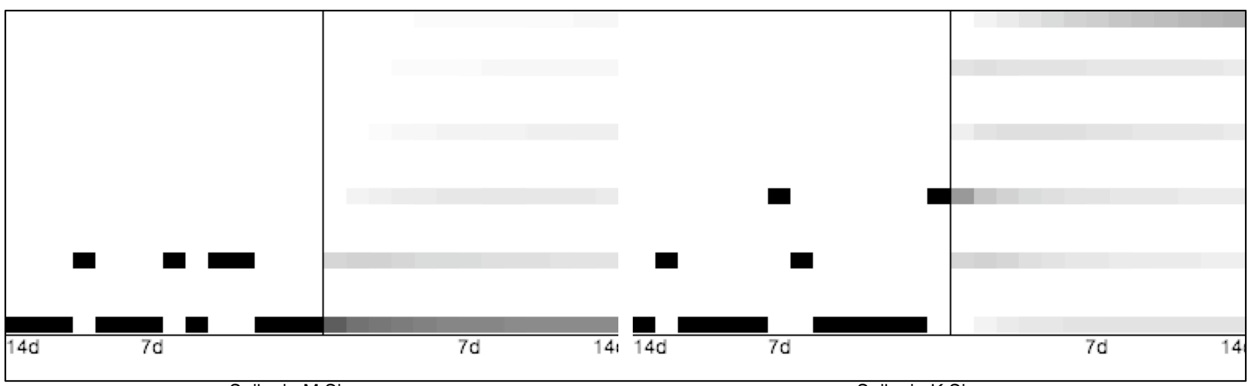

Spike in C Signups
Spike in F Signups


Spike in M Signups
Spike in K Signups



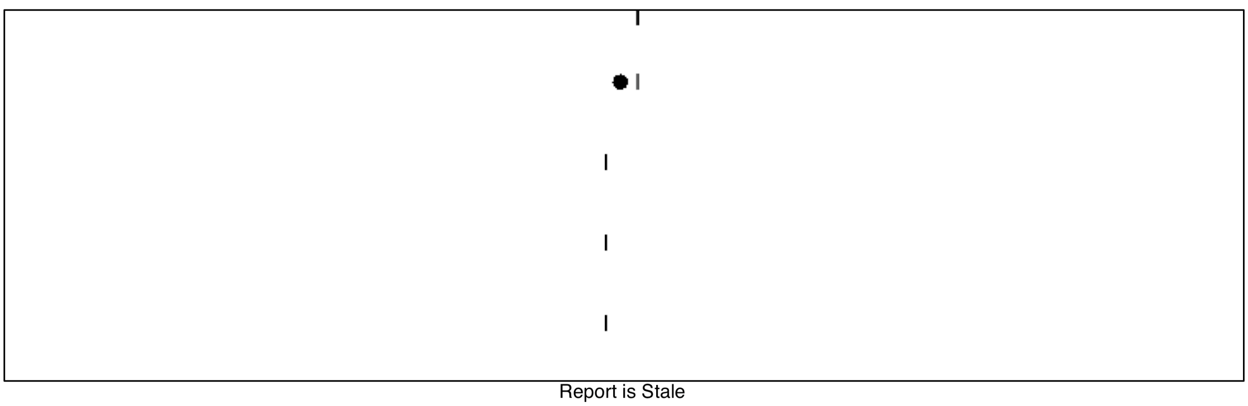
Spike in C Signups

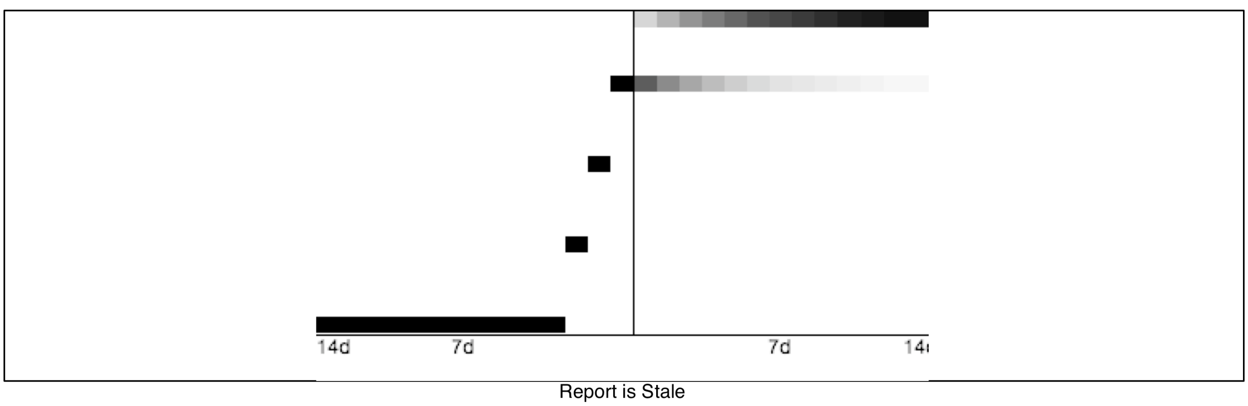
Spike in F Signups



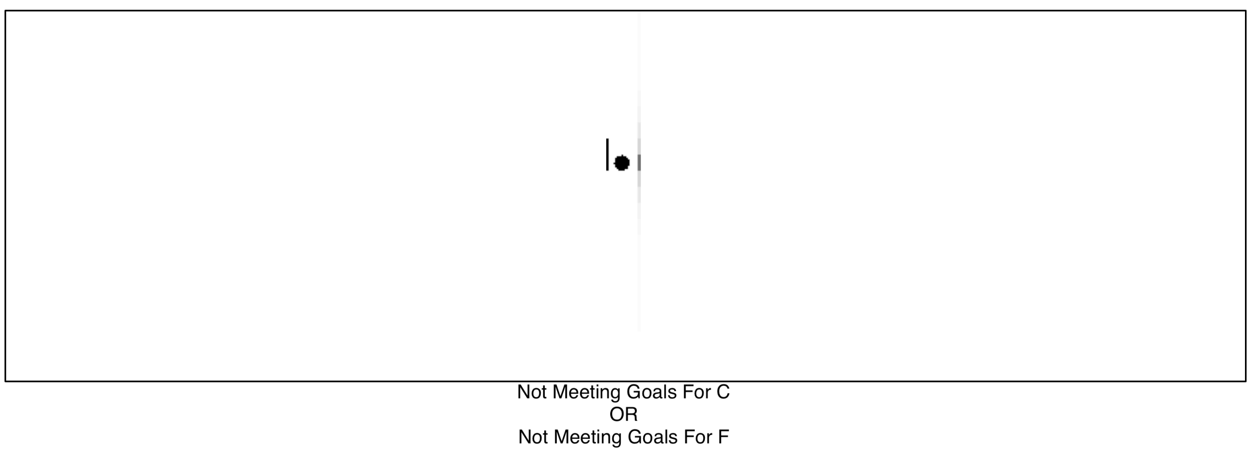
Spike in M Signups

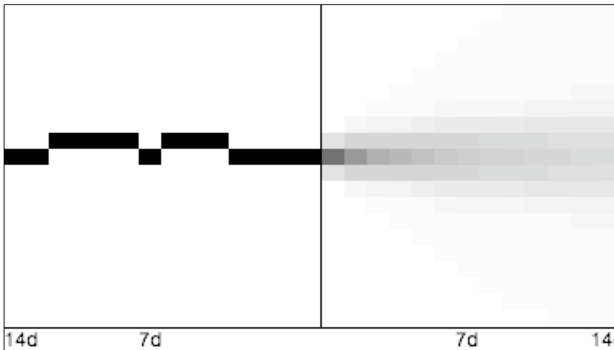
Spike in K Signups


Report is Stale

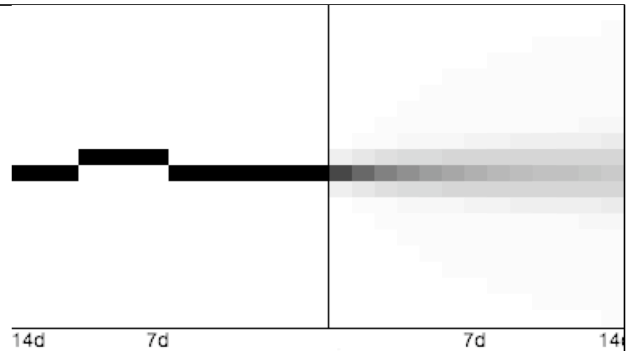

14d
7d
7d
14
Report is Stale


Not Meeting Goals For C
Not Meeting Goals For F

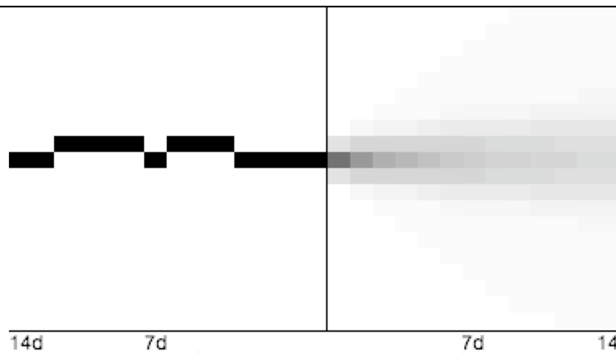

Not Meeting Goals For C
OR
Not Meeting Goals For F



Not Meeting Goals For C

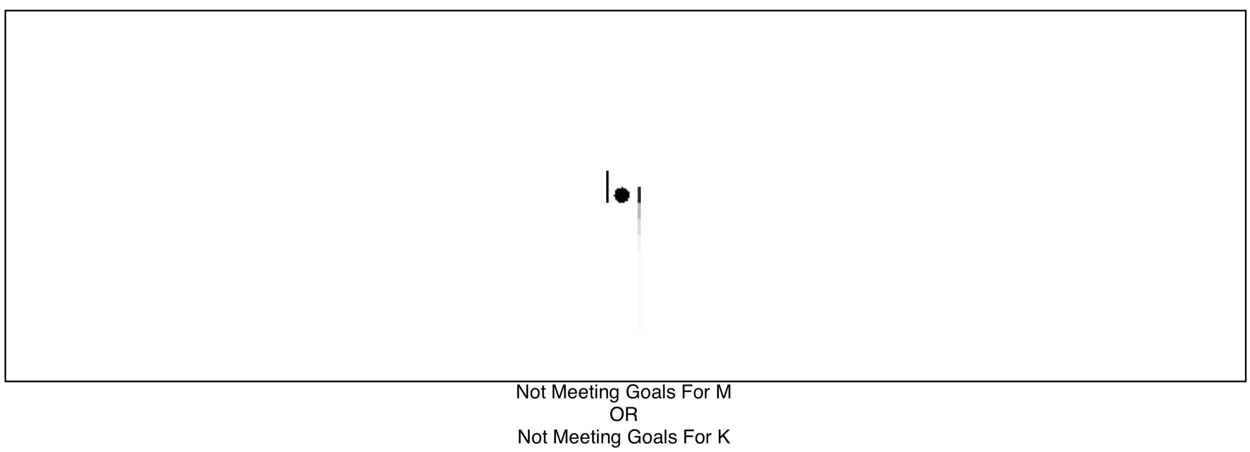


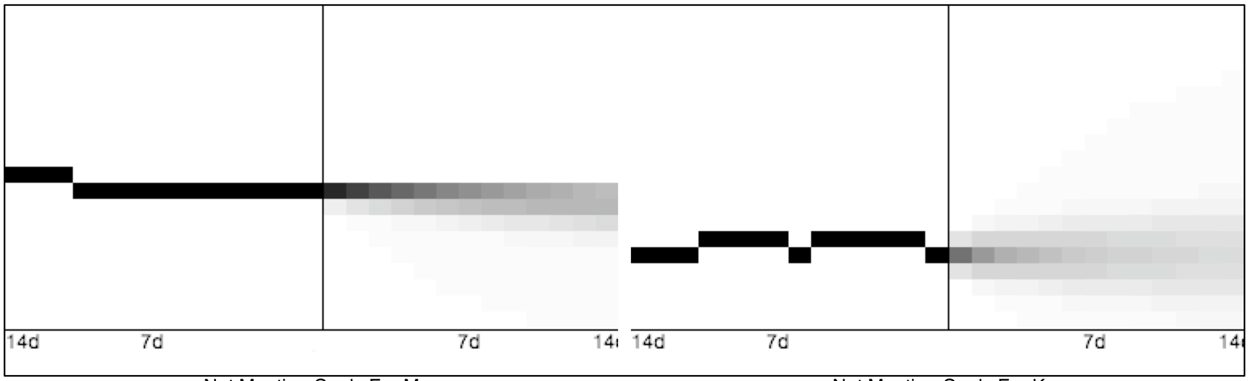
Not Meeting Goals For F



Not Meeting Goals For C
OR
Not Meeting Goals For F

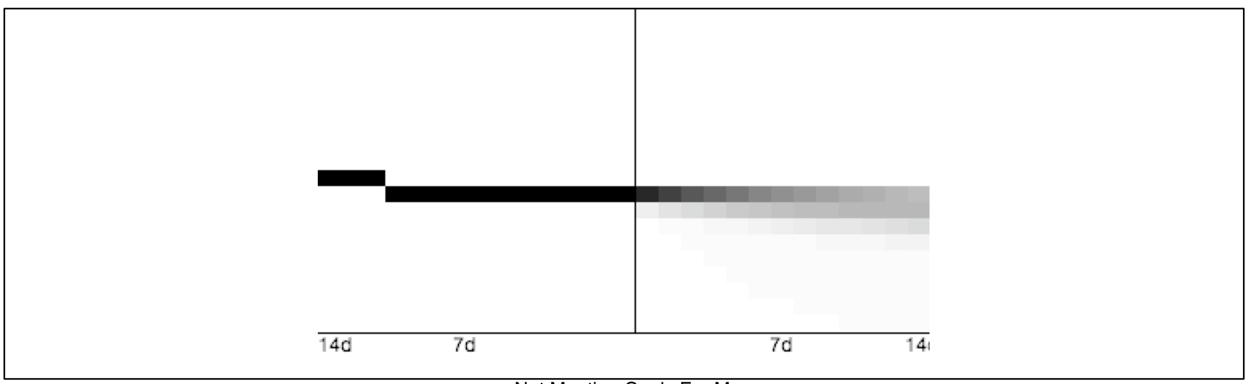

Not Meeting Goals For M
Not Meeting Goals For K


Not Meeting Goals For M
OR
Not Meeting Goals For K



Not Meeting Goals For M

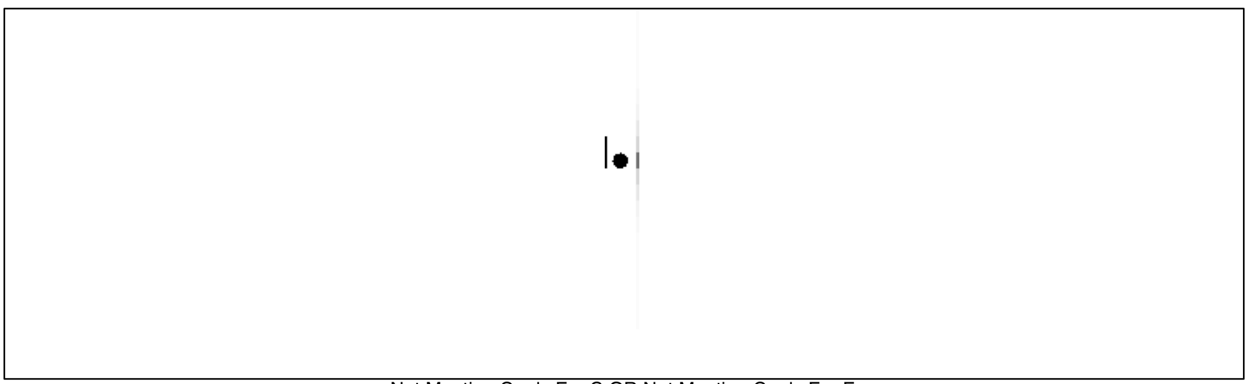
Not Meeting Goals For K



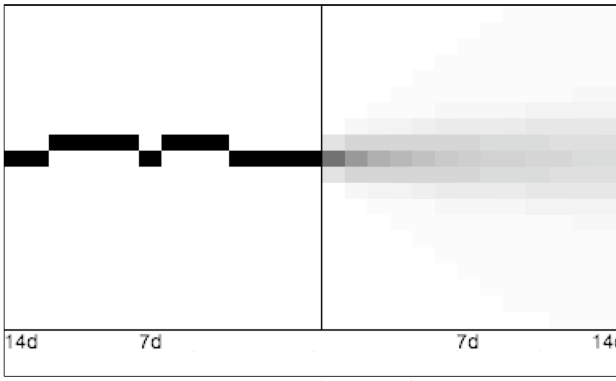
Not Meeting Goals For M
OR
Not Meeting Goals For K

Not Meeting Goals For C
OR
Not Meeting Goals For F

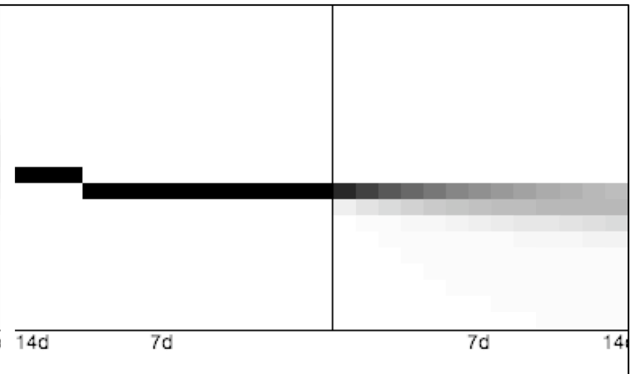
Not Meeting Goals For M
OR
Not Meeting Goals For K



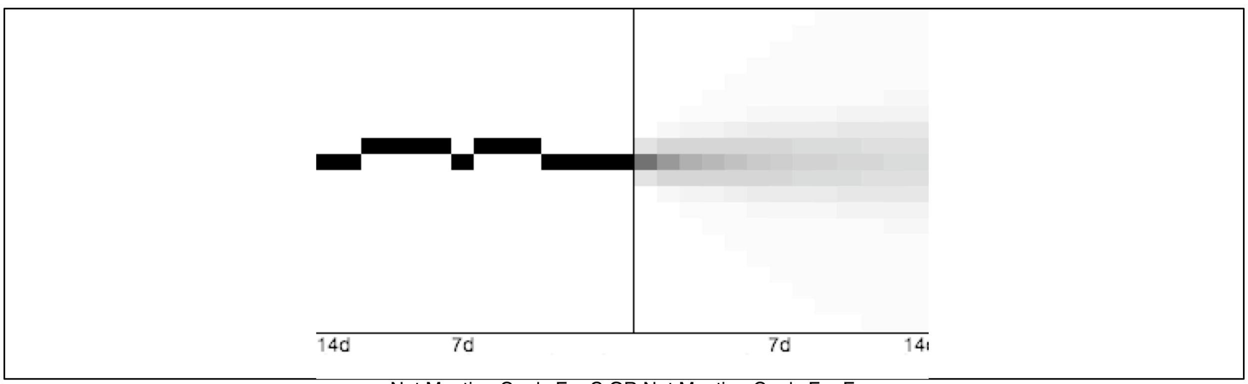
Not Meeting Goals For C OR Not Meeting Goals For F
OR
Not Meeting Goals For M OR Not Meeting Goals For K



Not Meeting Goals For C
OR
Not Meeting Goals For F

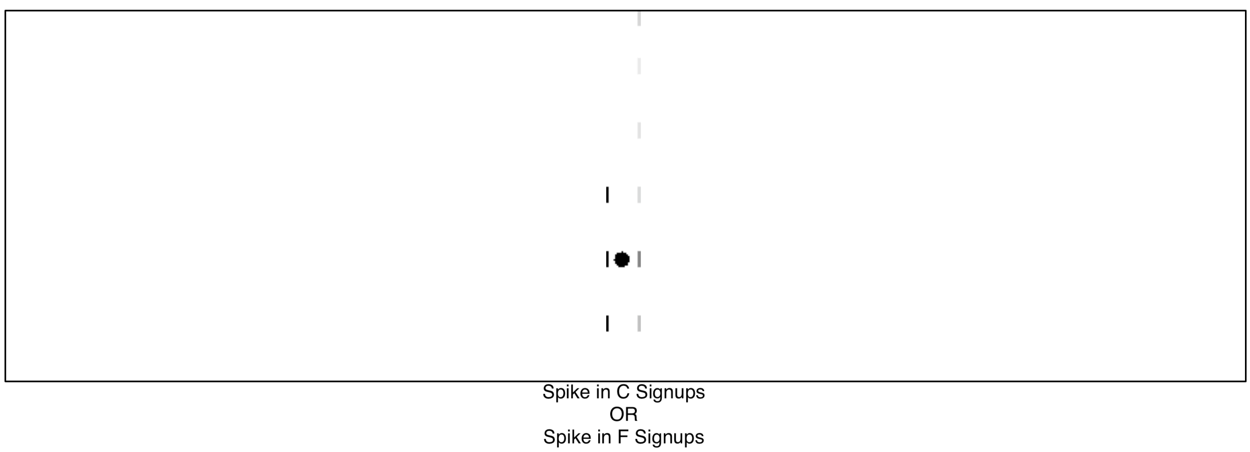


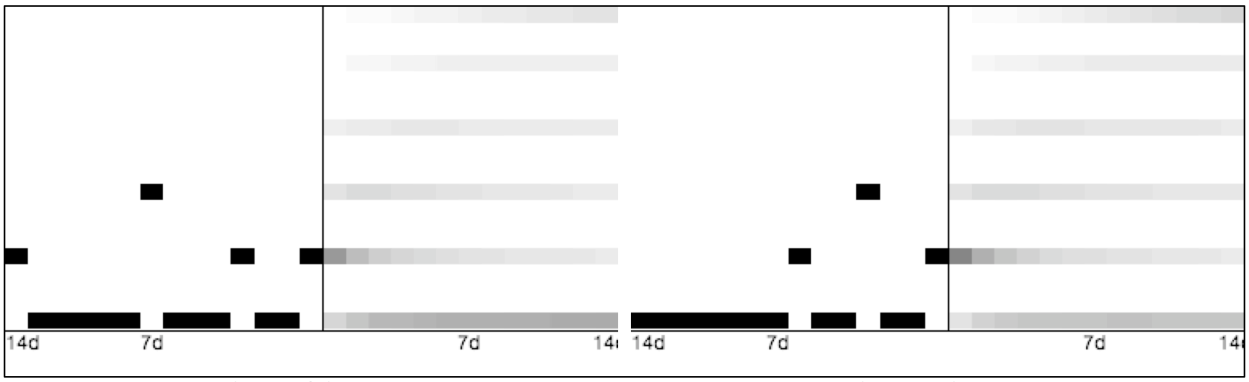
Not Meeting Goals For M
OR
Not Meeting Goals For K



Not Meeting Goals For C OR Not Meeting Goals For F
OR
Not Meeting Goals For M OR Not Meeting Goals For K

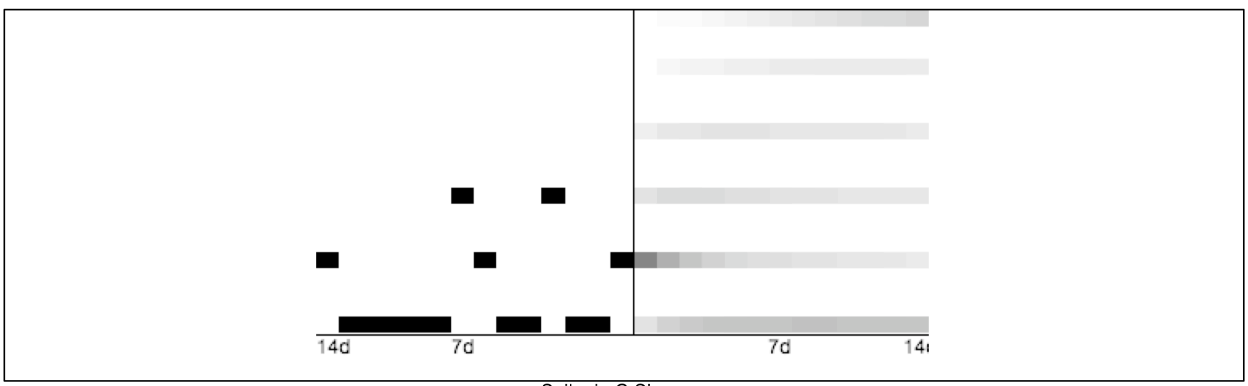

Spike in C Signups
Spike in F Signups


Spike in C Signups
OR
Spike in F Signups



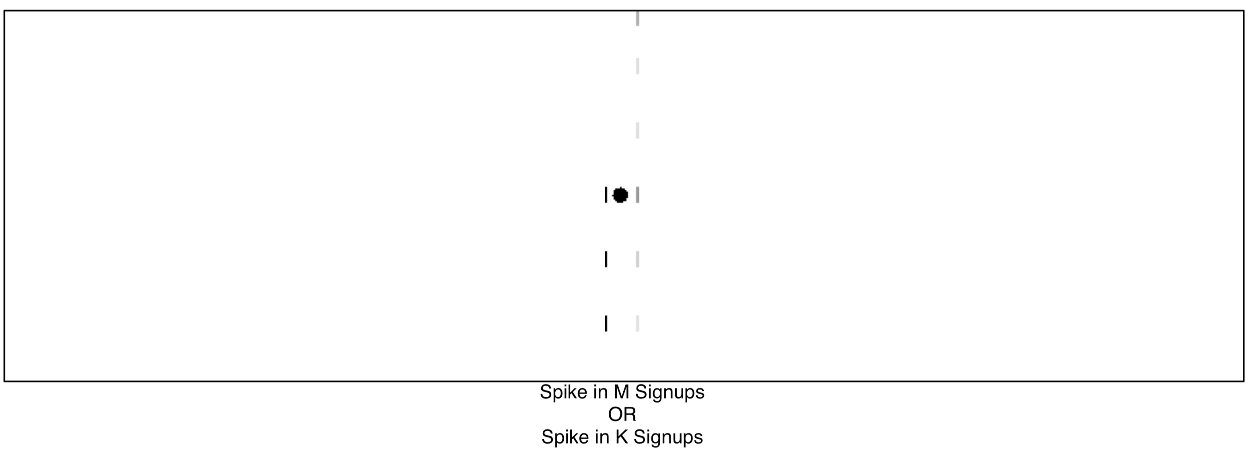
Spike in C Signups

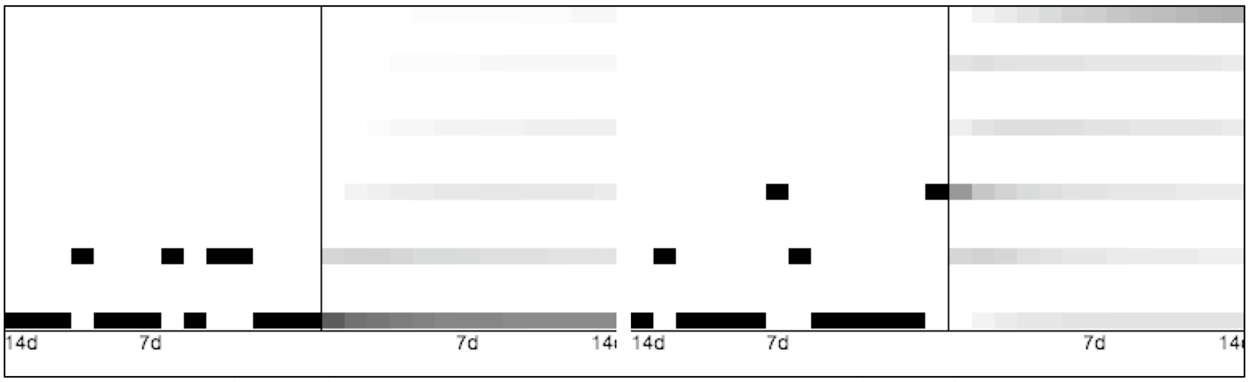
Spike in F Signups



Spike in C Signups
OR
Spike in F Signups

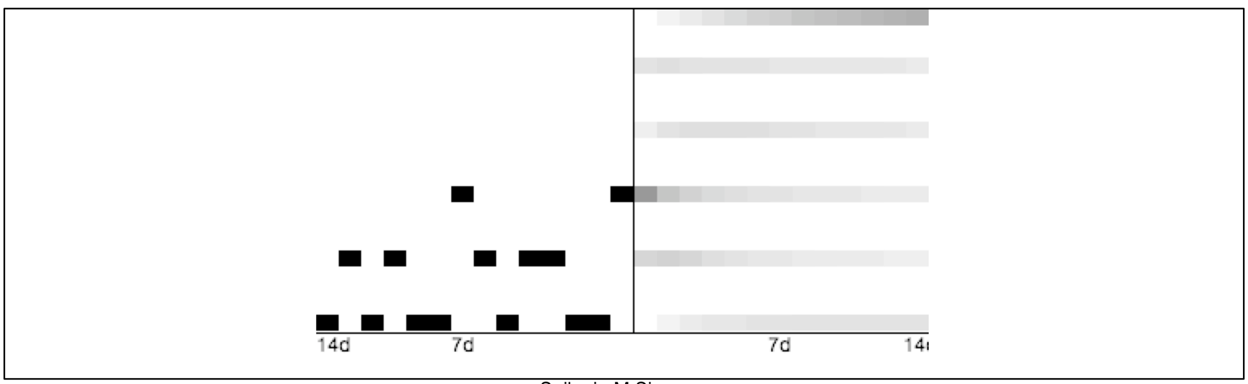

Spike in M Signups
Spike in K Signups


Spike in M Signups
OR
Spike in K Signups

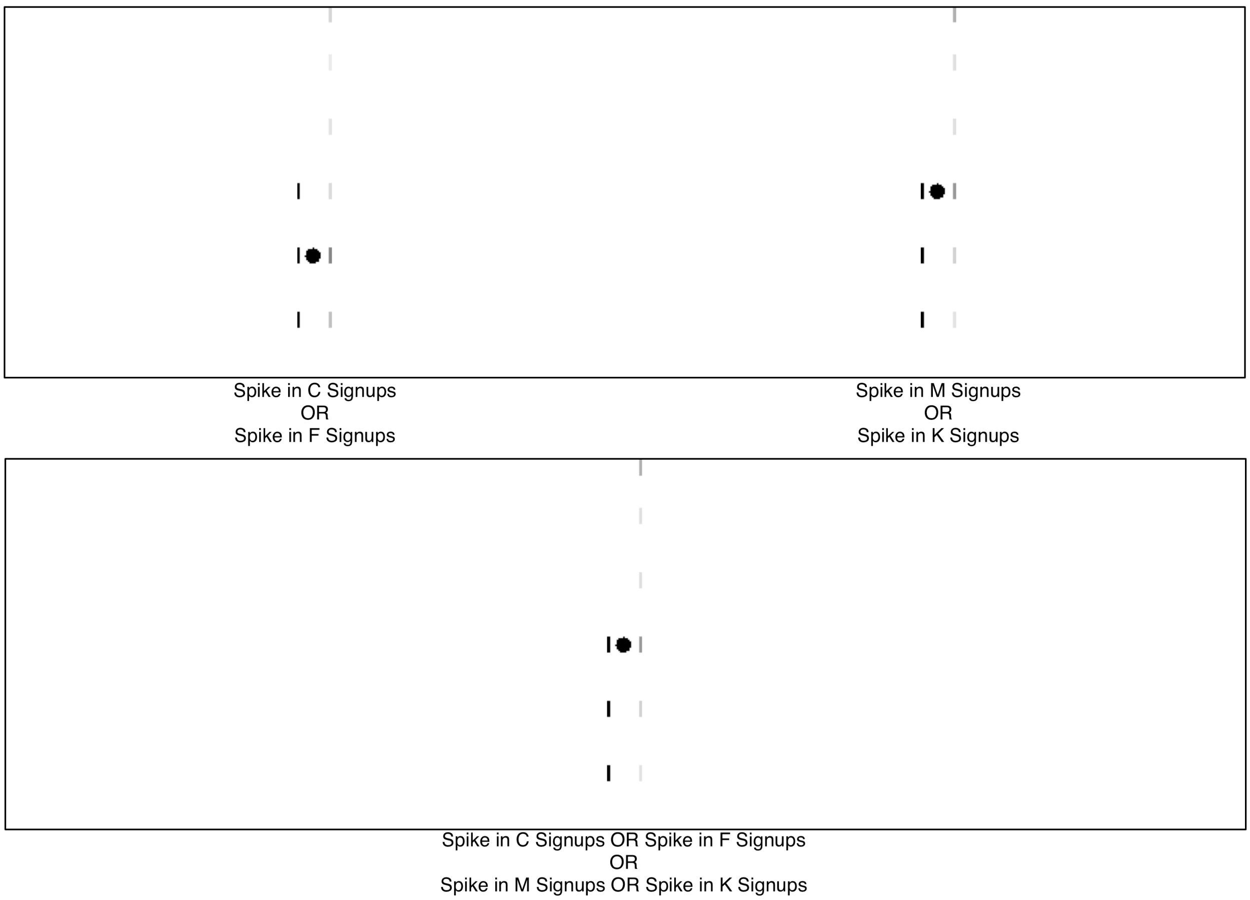


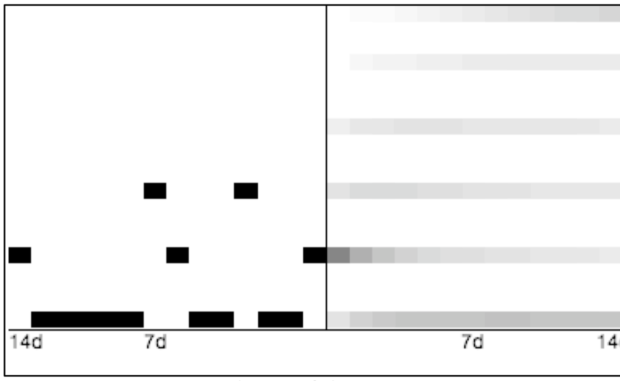
Spike in M Signups

Spike in K Signups

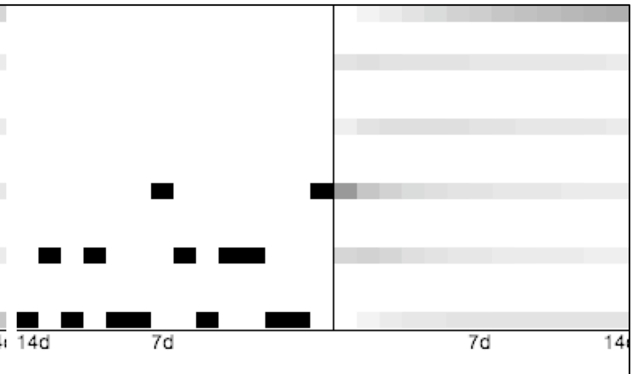


Spike in M Signups
OR
Spike in K Signups

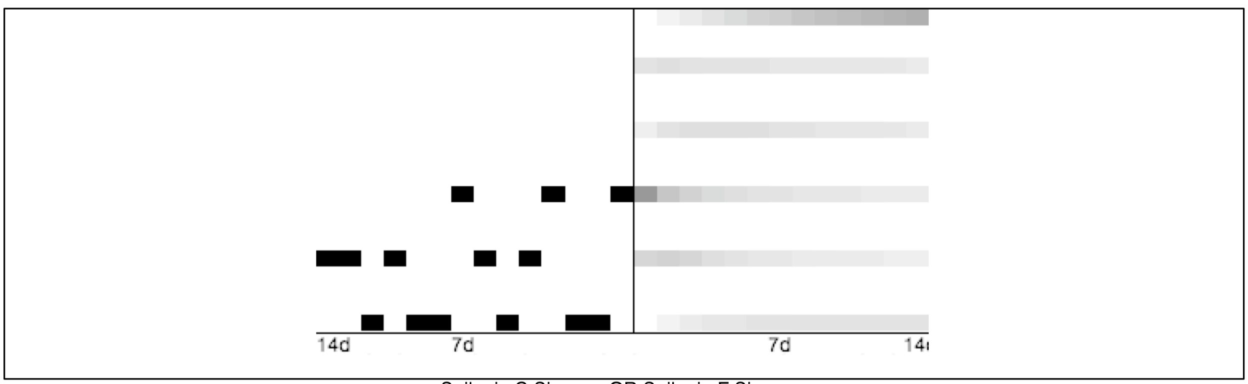

Spike in C Signups
OR
Spike in F Signups
Spike in M Signups
OR
Spike in K Signups
Spike in C Signups OR Spike in F Signups
OR
Spike in M Signups OR Spike in K Signups



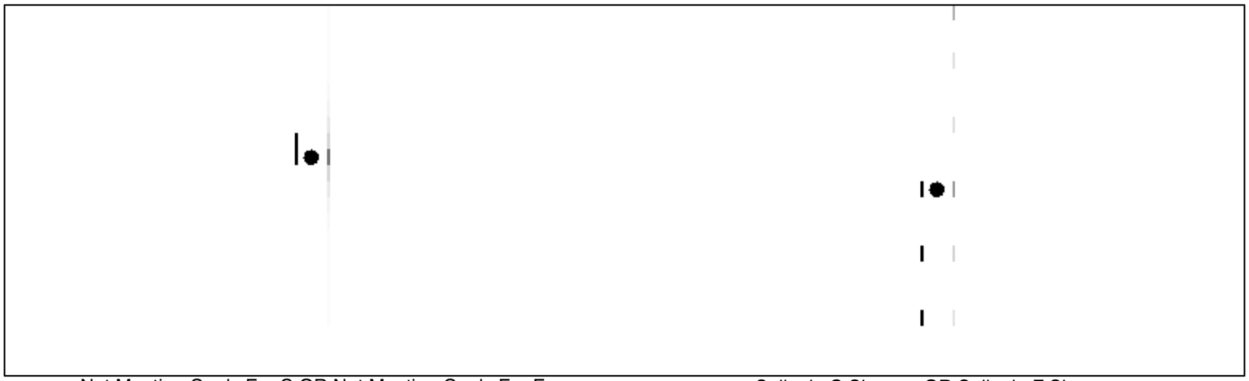
Spike in C Signups
OR
Spike in F Signups



Spike in M Signups
OR
Spike in K Signups

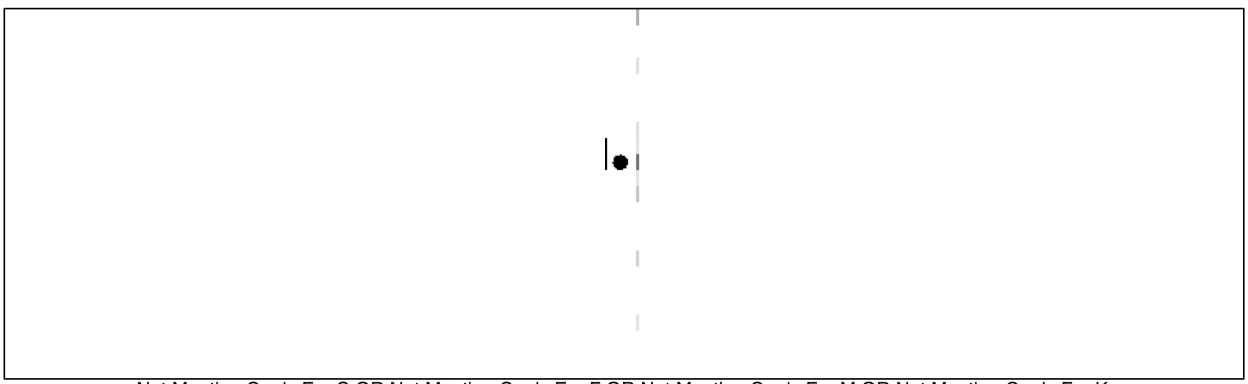


Spike in C Signups OR Spike in F Signups
OR
Spike in M Signups OR Spike in K Signups

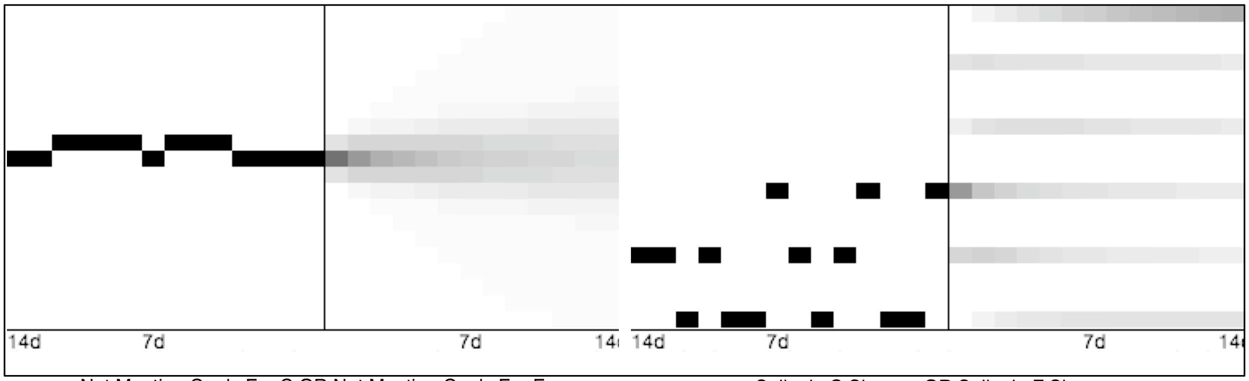


Not Meeting Goals For C OR Not Meeting Goals For F
OR
Not Meeting Goals For M OR Not Meeting Goals For K

Spike in C Signups OR Spike in F Signups
OR
Spike in M Signups OR Spike in K Signups

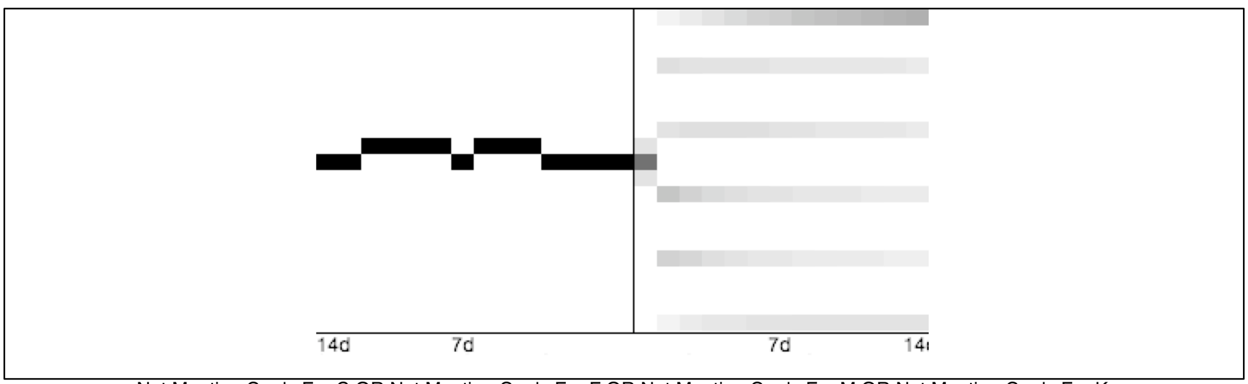


Not Meeting Goals For C OR Not Meeting Goals For F OR Not Meeting Goals For M OR Not Meeting Goals For K
OR
Spike in C Signups OR Spike in F Signups OR Spike in M Signups OR Spike in K Signups

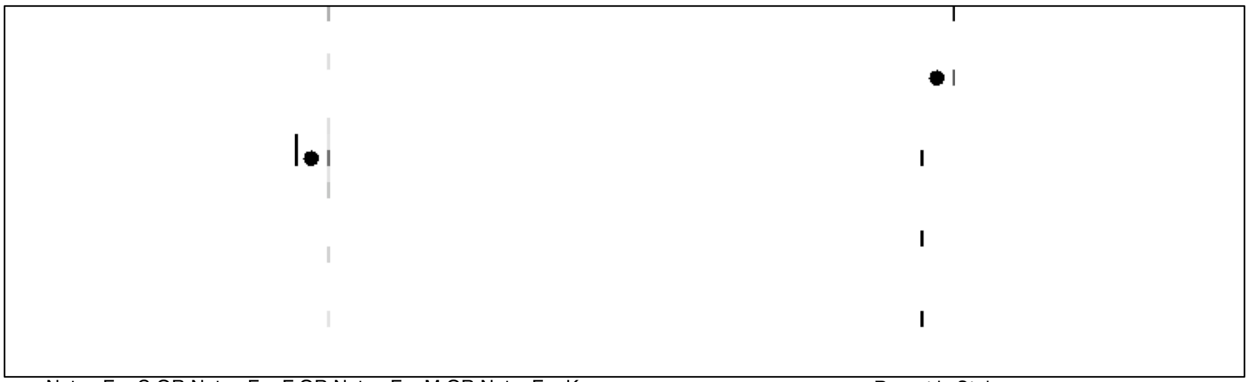


Not Meeting Goals For C OR Not Meeting Goals For F
OR
Not Meeting Goals For M OR Not Meeting Goals For K

Spike in C Signups OR Spike in F Signups
OR
Spike in M Signups OR Spike in K Signups

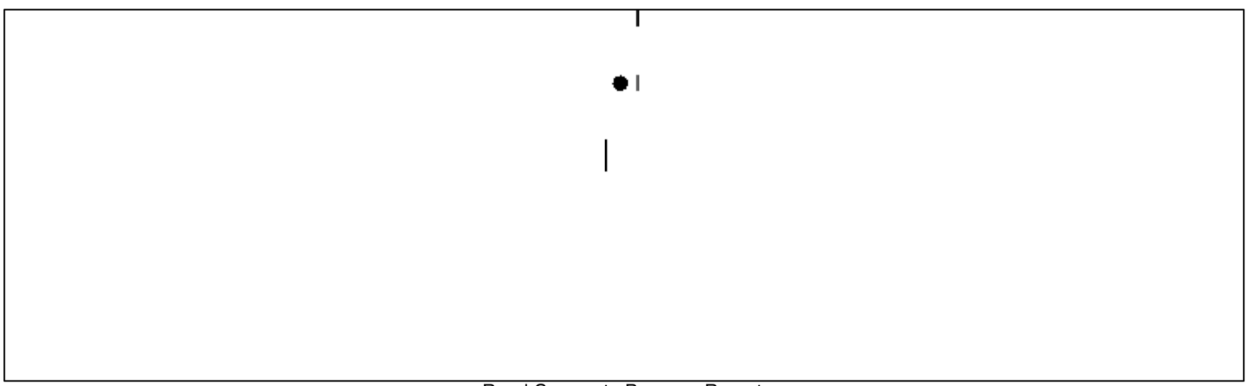


Not Meeting Goals For C OR Not Meeting Goals For F OR Not Meeting Goals For M OR Not Meeting Goals For K
OR
Spike in C Signups OR Spike in F Signups OR Spike in M Signups OR Spike in K Signups

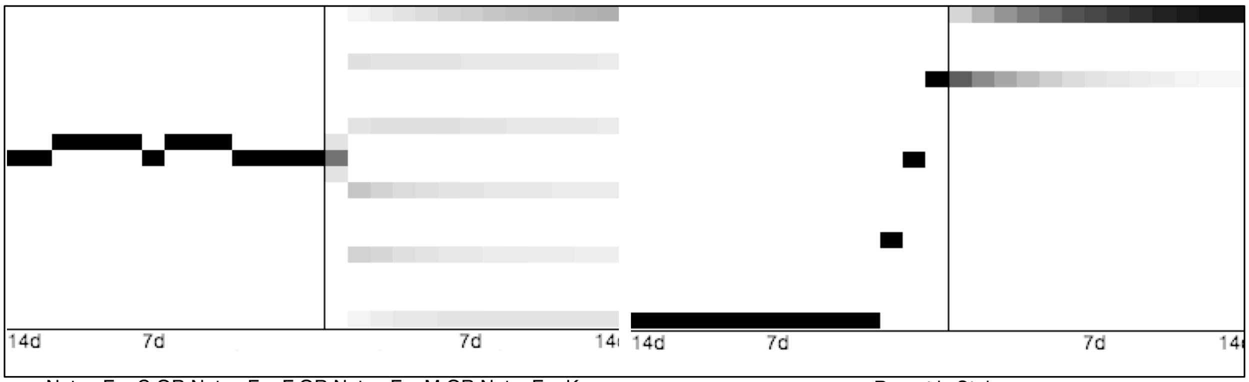


Not ... For C OR Not ... For F OR Not ... For M OR Not... For K
OR
Spike in C ... OR Spike in F ... OR Spike in M ... OR Spike in K ...

Report is Stale

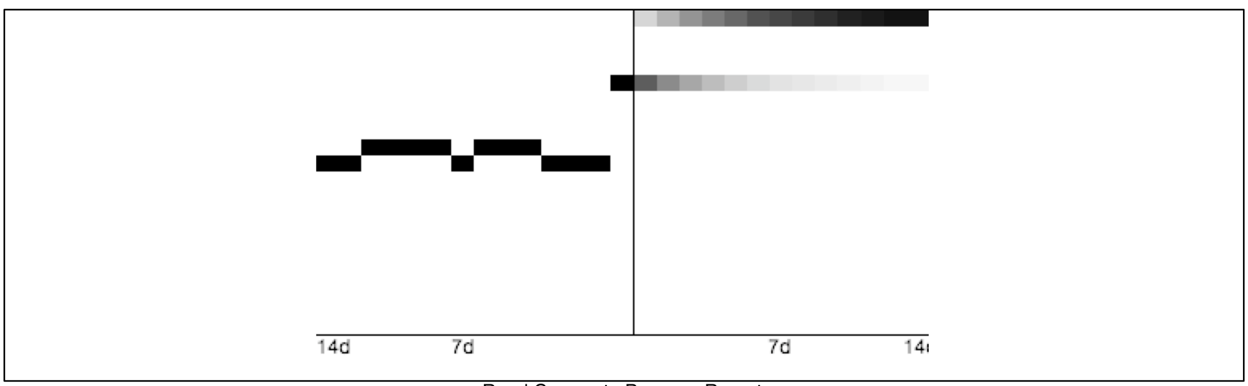


Read Corporate Program Report



Not ... For C OR Not ... For F OR Not ... For M OR Not... For K
OR
Spike in C ... OR Spike in F ... OR Spike in M ... OR Spike in K ...

Report is Stale



Read Corporate Program Report