Cognitive Work *of* Hypothesis Exploration *during* Anomaly Response

MARISA R. GRAYSON

A LOOK AT HOW WE RESPOND TO THE UNEXPECTED

Web-production software systems operate at an unprecedented scale today, requiring extensive automation to develop and maintain services. The systems are designed to adapt regularly to dynamic load to avoid the consequences of overloading portions of the network. As the software systems scale and complexity grows, it becomes more difficult to observe, model, and track how the systems function and malfunction. Anomalies inevitably arise, challenging incident responders to recognize and understand unusual behaviors as they plan and execute interventions to mitigate or resolve the threat of service outage. This is *anomaly response*.¹

The cognitive work of anomaly response has been studied in energy systems, space systems, and anesthetic management during surgery.^{9,10} Recently, it has been recognized as an essential part of managing web-production software systems. Web operations also provide the potential for new insights because all data about an incident response in a purely digital system is available, in

principle, to support detailed analysis. More importantly, the scale, autonomous capabilities, and complexity of web operations go well beyond the settings previously studied.^{7,8}

Four incidents from web-based software companies reveal important aspects of anomaly response processes when incidents arise in web operations, two of which are discussed in this article. One particular cognitive function examined in detail is hypothesis generation and exploration, given the impact of obscure automation on engineers' development of coherent models of the systems they manage. Each case was analyzed using the techniques and concepts of cognitive systems engineering.^{9,10} The set of cases provides a window into the cognitive work "above the line" [see "Above the Line, Below the Line" by Richard Cook in this issue] in incident management of complex web-operation systems [cf. Grayson, 2018].

THE REAL NATURE OF COGNITIVE WORK

It seems easy to look back at an incident and determine what went wrong. The difficulty is understanding what actually happened and how to learn from it. Hindsight bias narrows the ability to learn as it leads an after-the-fact review to oversimplify the situation that people faced and miss the real difficulties. When web requests are failing and customers cannot access content, however, people ask questions about what is malfunctioning, what is the underlying problem driving the disturbances observed, what interventions will mitigate or resolve the problems being experienced, etc.

Software engineering consists of sense-making in a

Understanding and resolving anomalies can require connecting experiences and knowledge gained from handling multiple past incidents.

highly dynamic environment with extensive and sometimes puzzling interdependencies in a network of systems mostly hidden “below the line.” Problems produce effects and disturbances that propagate and appear distant from the source driving behavior—effects at a distance in highly interdependent processes.⁶ Observing and tracing the behaviors of multiple automated processes and subsystems is difficult and ambiguous.

The people engaged in resolving the incident bring mental models about how the different components, functions, and subsystems are interconnected and update these models as they explore possible explanations. Understanding and resolving anomalies can require connecting experiences and knowledge gained from handling multiple past incidents. But no models of how the system works are identical or complete, so understanding the event requires work to integrate information and knowledge across the diverse perspectives. Hypothesis exploration and planning interventions are collaborative processes across distributed parties that could be all around the world [see “Managing the Hidden Costs of Coordination” by Laura Maguire in this issue].

WHAT IS ANOMALY RESPONSE AND HYPOTHESIS EXPLORATION?

Anomalies come in many forms, though the cognitive work of responding to them involves a basic set of key functions. An anomaly has two qualities: it is abnormal and unexpected, such as strangely slow response times for loading a home page or high network demand during a typically low traffic period in the middle of the night.

Recognizing an anomaly as a discrepancy between what is observed and what is expected of system behaviors depends on the observer's model of what the system is doing in that context. Anomalies are events that require explanation, since the current model of the system does not fit what is observed.

In other words, anomalies are triggers to generate and explore hypotheses about what is going on that would, if true, account for the anomalous behaviors.⁹ Multiple anomalies can build up over time as problems propagate through highly interdependent networks and as actions are taken to counter abnormal behaviors. Anomalies become an unfolding set of unexpected findings matched by generating an unfolding set of candidate hypotheses to test as potential explanations.

The cognitive work of anomaly response involves three interdependent lines of activity: (1) anomaly recognition, in which practitioners collect and update the set of findings to be explained; (2) hypothesis exploration, in which practitioners generate, revise, and test potential explanations that would account for the findings; and (3) response management or replanning, in which practitioners modify plans in progress to maintain system integrity, mitigate effects on critical goals, and determine what interventions can resolve the situation. Each of these is time-dependent and requires revision as evidence about anomalies and their driving sources comes in over time, remedial actions are taken that produce additional surprising effects, and pressure to resolve the situation builds even when uncertainty persists.

Given a set of findings to be explained, hypothesis

exploration generates and tests candidates. Interestingly, research shows that the difficulty of hypothesis exploration increases as the scale of interdependencies increases. In hypothesis generation, the goal is to broaden the set of hypotheses under consideration as candidate explanations for the pattern of findings and avoid premature narrowing. Research strongly suggests that diverse perspectives—in the right collaborative interplay—help generate a broader set of candidates. New events will occur while hypothesis exploration is under way, and these may reinforce current working hypotheses, elaborate on the current set of possible hypotheses, or overturn the current working hypothesis. The incredible speed of automation can quickly change the information landscape, further complicating hypothesis exploration.

ANOMALY RESPONSE CASES IN WEB OPERATIONS

The study under discussion here is based on four cases sampled from the larger corpus of available cases in the database developed by the SNAFUcatchers Consortium (<https://www.snafucatchers.com/about-us>), a group of industry leaders and researchers focused on understanding how people cope with complexity and produce resilient performance in the operation of critical digital services. The definition of *incident* varies among organizations, though most capture circumstances around service losses or degradations (e.g., Huang et al.³). The relevant parties and company-specific details were de-identified for the analysis.

Chat-log files were gathered from postmortem records as the primary data source for each of the cases. The

chat logs were either from IRC (Internet Relay Chat) or Slack, depending on the main communication technology used at the time. The chat logs do not directly show the actions of the engineers on the system; they do record intention and plans that individuals expressed above the line in the process of responding to anomalies and the noted signals crossing the line.¹¹ Additionally, the chat operations demonstrate the emergence of anomalies in the observer's implied stance, given the written updates in the main channels. The data records were supplemented with knowledge-elicitation sessions with individuals who had direct knowledge of the incidents.

The analysis used process-tracing methods.⁵ Over several iterations, the communication logs were analyzed by applying a lightweight coding scheme based on the cognitive work of anomaly response and macrocognitive functions.^{4,6} The focus was on several key processes, including (1) events; (2) hypothesis generation; (3) model revisions; (4) interventions; and (5) stance.² These five aspects captured the expectations and communication flow of the engineers responding to cascading disturbances.

This paper focuses on the results of hypothesis generation and exploration. The engineers communicate active theories to provide direction for diagnostic search, as well as broadening the hypothesis-exploration space with contributions from multiple perspectives. The evolution of the hypothesis space is marked and laid out diagrammatically (figures 1-3), featuring activities such as adding or ruling out hypotheses; findings that support active hypotheses; hypothesis modifications; revisiting

past hypotheses; mental model updates; and points of confusion and uncertainty.

VISUALIZING THE HYPOTHESIS EXPLORATION SPACE

Over the course of the cases, software engineers offered many hypotheses. The anomalous signs and signals prompted new ideas to emerge, as well as supporting the evolution or dismissal of other explanations. The chat channels enabled open communication about these hypotheses in a collective landscape that was theoretically available to all participants at any time during the incident. The parallel cognitive paths were laid out for each case to show the diverse patterns of action and insight brought to bear.

The top portions of the diagrams, exemplified in figure 1, portray the different hypotheses in the shared landscape. Each bubble has a condensed version of evidence and proffered conclusion. The hypothesis-exploration space (top portion) is marked with various hypothesis bubbles, which are supported by the anomalous signs (middle) and the shared interventions and results (bottom). The line of commitment separates the hypotheses that were acted upon, though may have been proven false or irrelevant depending on the case. The hypotheses are connected, showing both divergence and convergence over time. Notably, some hypotheses were ultimately dismissed (red outline at minutes 15 and 50) or noted as irrelevant to the matter at hand (black outline at minutes 50, 60, and 70). The line of commitment marks a point where action was taken, often in spite of uncertainty.

[click for figure 1]

The middle portion of each figure supports the upper section with specific moments of anomalous signs and signals. Each marker denotes the time since the incident's start and anomalous state. The bottom portion shows interventions and clarifying questions the engineers made during the incident. These actions could be diagnostic or therapeutic depending on the case.

Both the signals and the interventions have arrows driving toward single or multiple hypotheses in a similar alignment to that of representations toward above-the-line practitioners in the "Above the Line, Below the Line" diagram shown in the article by Cook in this issue. The hypotheses are generated above the line and are motivated from the anomalies and interactions arising from the line of representation.

Next let's examine two of the case studies, in which the narrative of investigation and mitigation is shown in a graphical timeline representation.

A CASE OF WIDESPREAD LATENCY

Background: two data centers house the databases and servers needed to run a website. Backups are periodically stored to protect the data if anything goes wrong and are kept separately within each data center. Network pipelines connect the two, as well as connecting them to the terminals that the software engineers use to sustain the site. Little do they know that what appears to be a reasonable minor change will have widespread consequences.

A search engineer is on call when an automated alert is triggered. He and another engineer gather several people

online in the predawn hours to diagnose the anomalous behavior in multiple systems. Increased latency, lag, and connection issues are rampant across search, memcache, monitoring charts, and the production website. The network connection issues are the driving source of the overload, though it is not immediately obvious to the search and operations engineers. They see the symptoms of the network degradation but have limited access to trace the underlying issues. Their initial hypotheses are tentative as they gather more information.

As seen in the top section in figure 2, early action is taken to switch the server groups. The hypothesis-exploration space (top portion) is supported by the anomalous signs (middle) noted by the engineers in the chat logs, as well as the shared interventions and results (bottom). The hypotheses eventually converge in this case to surpass the relative line to commit to a plan of action. Often a problem with one half would not affect the nearly identical other half. That hypothesis is quickly abandoned, however, to follow other ideas about what is driving the widespread disturbances.

[click for figure 2]

The top portion of figure 2 shows the evolution of the engineers' hypotheses as they move from one idea to the next. The red ones (dark-outlined hypotheses around minutes 15 and 50 in the timeline) are disproven, and the black are deemed irrelevant to the issue at hand. The two sections below note instances of anomalous behavior and interventions with shared results, respectively. Both

Both groups provide evidence and context for the different hypotheses proposed over the course of the event. At the same time, the other engineers debate whether a network issue is the actual cause of the slowdown.

groups provide evidence and context for the different hypotheses proposed over the course of the event. For example, 30 minutes have passed, and a few hypotheses about the driving problem form. Several connection errors to search and memcache lead a few engineers to conclude that memcache's performance is instigating the other issues. At the same time, the other engineers debate whether a network issue is the actual cause of the slowdown.

The effects are widespread and at a distance from the real problem with no discernible connection. Eventually, the engineers track down the deeper source of the issue, and their conclusions are supported by a network engineer with greater access to the desired metrics. He dismisses the memcache theory and explains that the network traffic is unusually high within the system. The bottleneck is the pipeline delivering the backups between the data centers, which also carries data vital to serving the site's other main functions. The effects manifest as overall slowness because there are almost no resources left to continuously facilitate functionality other than the backups on the site. The backup process fluctuates enough to allow some bandwidth to be used by other areas, but overall the latency is very high across the functions. The need for other parts of the system to access the servers through the same pathway increases the overload of the pipeline, which effectively breaks the system.

The relay cards delivering the network data across data centers have no throttle to prevent their resource capability from being maxed out. The conduit is normally oversubscribed, meaning that the inputs could overload

the pathway multiple times over if used at capacity. The resource utilization is typically much lower than the theoretical threshold, so the risk is traded off for greater accessibility over a common route. A physical analogy for this case is a pipe that usually has a small amount of water flowing through it, now being flooded all at once. In this case, the other functions are starved for bandwidth and cannot adapt to the lack of resources.

About an hour in, one engineer recalls a recent change to the backup structure after the errors mysteriously went away. By this point, the hypotheses have pushed past the line of commitment as the engineers have decided on a course of action. Earlier in the week, the backup process had been altered to go across the two data centers rather than staying within each. Essentially, the servers are now sending data to the backup servers in the other data center instead of those housed in the same data center. Since database backups usually occur weekly, the system's performance was not impacted until the backup process was initiated for the week. The new perspective and evidence from the network engineer cut through the uncertainty to establish a clear assessment of the source of the trouble and provide the course of action for resolution—reverting to the old backup scheme until they can develop one that will not overload the system.

A CASE OF DISAPPEARING REQUESTS

This setup may sound familiar: A group of servers including databases and APIs are managed by a load balancer. As the name implies, the load balancer dynamically distributes load or requests across the servers that it oversees to

avoid overloading any particular asset. Normally, this process works without interference, except this time.

The first warning signs come from a custom email alert by one product engineer that there are connection errors to the API servers. The errors are intermittent and hard to reproduce consistently among multiple individuals. The normal alerting thresholds are not triggered, so the product engineer alerts other knowledgeable engineers in operations and infrastructure to assist. No errors are appearing in the typical logging monitors, which seems very strange to the responders. They determine the connections are not reaching the servers but cannot support their hypotheses about where the problem resides. One person recalls that a recent teeing junction was put in place to test certain boxes before adding them to production by diverting a small amount of traffic to them.

As shown in figure 3, the software engineers decide to remove the testing boxes early on, which does not resolve the issue. The shared hypothesis space (top) in this case features several divergent hypotheses, some of which are acted upon and eventually combined. The anomalies (middle) are fairly consistent throughout the incident, while the interventions (bottom) deal with questioning and following different lines of inquiry.

[click for figure 3]

Many hypotheses are discussed and available to the participants in the chat, though it takes revisiting old ideas and integrating new ones to form any satisfactory

conclusions. Although the participants later realize that some hypotheses were incorrect, the information they discover by following several paths of inquiry evolve into other hypotheses.

The initial responders have limited access to the load-balancer logs and are mainly focused on the application-level components, without much success. A network engineer with load-balancer access is called in to check on the connections and finds an old rule on the same cluster of servers in question. After removing the rule and the most recently added teeing rule, the engineer finds that the errors disappear. Together with the infrastructural engineer who implemented the teeing rule, they determine that the rules had unexpectedly interacted and reactivated the old one, which directed requests to a box that no longer existed.

The basic assumption for the load-balancer structure is that it will not send traffic to a server that is not able to handle it. It accomplishes this purpose with a health check, a short request response verifying that the server has available capacity. The teeing rules, however, do not automatically have this health check and can send requests to a box that might not exist. Furthermore, the interaction of multiple rules could have some influence on this check if one set has a valid target while the other does not. It is hard for the engineers to estimate the downstream impact of the dropped requests for their system's functions and the end users' experience since the fraction of diverted requests is so small and invisible to monitoring.

Prior testing rules were not actively curated and

Eventually, a few theories emerge as plausible explanations for the apparent zombie rule, although no definitive consensus is reached without further testing.

are left on the machines without overt influence over the network traffic flow. A normal testing structure is added that inadvertently reactivates an older junction rule that instead sends traffic to a box that has been decommissioned. After this discovery is made, there is still confusion as to why the new rule interacted with the old one when they should have been independent. Hypotheses are acted upon without great certainty as to the effects, such as removing the teeing rules completely. The network engineer with the most access and directability on the load balancer also struggles to understand the entanglements that triggered the sinkhole of dropped requests. Eventually, a few theories emerge as plausible explanations for the apparent zombie rule, although no definitive consensus is reached without further testing.

EXPLORING NEW HYPOTHESIS LANDSCAPES

The groups of engineers in each of these cases explored their hypothesis spaces in different ways, though both had common challenges in reaching their incidents' "conclusions." The true nature of incidents is in the continual flow of day-to-day operations rather than the short duration captured in a postmortem. Nevertheless, the captured cases do show relevant patterns such as exploring both narrowly and more broadly.

The first case saw the ideas converging toward a fairly confident plan of action. In contrast, the second case had hypotheses that the engineers committed to early on and many divergent paths without a clear resolution. Both had initial responses that were unsuccessful in accomplishing the desired result, but provided additional information to

direct subsequent hypotheses. Each probe into the system and search spurred new ideas to be added to the collective hypothesis space.

The engineers in both cases demonstrated a vital skill of interpreting data and providing context to the ambiguous signals. The underlying automation is opaque, especially when performing highly autonomous functions such as distributing network traffic in a load balancer. Effects emerging at a distance from their sources, however, presented in highly interpretable ways. Each case demonstrated a different set of signals observable to the engineers over the course of the incident. Another side effect of the interdependent, opaque network is masking, which obscures the automation-driven functions that might be relevant. The diversity of pathways through which overload can occur and surface is a symptom of the complexity of the network, which requires deeper and more informative measures for investigation.

Hypothesis exploration is complicated by the interacting effects hidden below the observable monitors. Limited measurable signals, masking, and strange loops restrict the human responders' abilities to understand the systems and take appropriate corrective actions. Time also affected the scope of investigation. Recent changes were given priority as likely contributors to the current issue, even when evidence may have supported other explanations.

It is much harder to trace changes disjointed in time, such as a week prior or long-term choices that left latent effects waiting to be activated by specific circumstances. One major difficulty in tracing anomalies in complex

software systems is the system's constant state of change. Hundreds of updates occur each day, varying in length, though their impacts might not be felt until much later as a cumulative effect. Current alerting platforms often provide localized information, which can help support focused hypotheses and also narrow the scope of investigation. The responders in these cases were hindered by the lack of observability into the underlying dynamics of the automation.

The tracings of the hypothesis-exploration space in this article specifically reveal unexpected patterns for incident management: (1) there are multiple committed hypotheses and interventions above the line; (2) many hypotheses are generated in a short time; (3) more actions and hypotheses are continually made after one was committed; and (4) the opaqueness and complexity of the highly dynamic systems produce effects at a distance that complicated hypothesis exploration.

The complexity of the system and autonomous functions drove investigators to collaborate and explore multiple hypotheses in responding to the anomalies. Diverse perspectives expanded the hypotheses considered and beneficially broadened the scope of investigation. Explicit comments by engineers updating each other's mental models were frequent in the chat logs; this finding supports Woods' theorem on the importance of finding and filling gaps in understanding.⁷

The chartings of hypothesis evolution also demonstrate the influence of a collective idea space via the communication channel. Early divergence of multiple hypotheses led to some tentative commitments to

action, as well as ruling out irrelevant contributions. The discarded ideas often helped other paths gain momentum toward a general convergence sufficiently explaining the anomaly. The unique experiences, skill sets, and roles of the individual responders contributed to resolving the complex challenges.

Related articles

➡ The Debugging Mindset
Understanding the psychology of learning strategies leads to effective problem-solving skills.
Devon H. O'Dell
<https://queue.acm.org/detail.cfm?id=3068754>

➡ Searching Vs. Finding
William A. Woods
Why systems need knowledge to find what you really want
<https://queue.acm.org/detail.cfm?id=988405>
➡ User Interface Designers, Slaves of Fashion
The status quo prevails in interface design, and the flawed concept of cut-and-paste is a perfect example.
Jef Raskin
<https://queue.acm.org/detail.cfm?id=945161>

Ultimately, sharing ideas and investigating several hypotheses broadened the engineers' views of the problem enough to find reasonable solutions. Whether it was a cumulative progression of evidence or eureka moments after finding the right monitoring source, the incident responders were able to intervene and protect the functionality of the software systems. High-reliability continuous development and deployment pressures engineers to keep pace with change and adapt to constant challenges. Their hypothesis exploration should be supported by the tools they use every day because they are

already solving problems that end users never even know about.

References

1. Allspaw, J. 2015. Trade-offs under pressure: heuristics and observations of teams resolving Internet service outages. Master's thesis. Lund, Sweden: Lund University.
2. Chow, R., Christoffersen, K., Woods, D. D. 2000. A model of communication in support of distributed anomaly response and replanning. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting 44* (1), 34-37. Sage Publications.
3. Huang, P., Guo, C., Zhou, L., Lorch, J. R., Dang, Y., Chintalapati, M., Yao, R. 2017. Gray failure: the Achilles' heel of cloud-scale systems. In *Proceedings of the 16th Workshop on Hot Topics in Operating Systems*, 150-155. ACM.
4. Klein, G., Ross, K. G., Moon, B. M., Klein, D. E., Hoffman, R. R., Hollnagel, E. 2003. Macrocognition. *IEEE Intelligent Systems* 18(3), 81-85.
5. Woods, D. D. 1993. Process tracing methods for the study of cognition outside of the experimental psychology laboratory. In *Decision making in Action: Models and Methods*, eds. G. A. Klein, J. Orasanu, R. Calderwood, C. E. Zsombok, 228-251. Westport, CT: Ablex Publishing.
6. Woods, D. D. 1994. Cognitive demands and activities in dynamic fault management: abductive reasoning and disturbance management. In *Human Factors in Alarm Design*, ed. N. A. Stanton, 63-92. Bristol, PA: Taylor & Francis.
7. Woods, D. D., ed. 2017. STELLA: Report from the SNAFUcatchers Workshop on Coping with Complexity; <https://snafucatchers.github.io>.

8. Woods, D.D. 2018. The strategic agility gap: how organizations are slow and stale to adapt in a turbulent world. In *Human and Organizational Factors in High-Risk Companies*, eds. F. Daniellou and R. Amalberti, Toulouse, France: Foundation for Industrial Safety Culture (FONCSI).
9. Woods, D. D., Hollnagel, E. 2006. Anomaly response. In *Joint Cognitive Systems: Patterns in Cognitive Systems Engineering*, 69-95. Boca Raton: CRC/Taylor & Francis.
10. Woods, D. D., Hollnagel, E. 2006. Automation surprise. In *Joint Cognitive Systems: Patterns in Cognitive Systems Engineering*, 113-142. Boca Raton: CRC/Taylor & Francis.
11. Woods, D. D., Patterson, E. S., Roth, E. M. 2002. Can we ever escape from data overload? A cognitive systems diagnosis. *Cognition, Technology & Work* 4(1), 22-36.

Marisa R. Grayson is an award-winning cognitive systems engineer at Mile Two, LLC. She holds a master's degree from the Ohio State University and is a member of the Cognitive Systems Engineering Lab and SNAFU Catchers Consortium. Her experience draws inspiration from research across multiple domains such as health care, defense intelligence, and distributed software systems. As an avid UI developer and gamer, she designs useful and usable products that show the narrative of real work.

Copyright © 2019 held by owner/author. Publication rights licensed to ACM.