Work Practices of System Administrators: Implications for Tool Design

Nicole F. Velasquez
IBM T.J. Watson Research Lab
19 Skyline Dr
Hawthorne, NY 10532
nicolefv@gmail.com

Suzanne P. Weisband
The University of Arizona
1130 E Helen Street
Tucson, AZ 85721
weisband@eller.arizona.edu

ABSTRACT
System administrators are specialized workers and computer users. As skilled workers in complex and high-risk environments, intuition tells us this unique user group may have requirements of the systems and software they use that differ from the requirements of regular computer users. An examination of system administrator work practices sheds light on the system attributes and characteristics they need to do their jobs. Through shadowing, interviews, and a review of previous system administrator studies, we present information and system quality attributes that appear to be important to system administrators. Following a discussion of these attributes, we present a model of user satisfaction that provides actionable guidance and an integration of the attributes. We close with a discussion of the research findings and a call for future research in this area.

Categories and Subject Descriptors
H.5.2 [**User Interfaces**]: *User-centered design, evaluation/methodology, interaction styles*, K.6.4 **[System Management]**.

General Terms
Design, Human Factors

Keywords
System administrator, work practice, design, user satisfaction

1. INTRODUCTION
System administrators (sysadmins) are becoming increasingly important as organizations continue to embrace technology. With responsibilities that can include the installation, configuration, monitoring, troubleshooting, and maintenance of increasingly complex and mission-critical systems, their work distinguishes them from everyday computer users, and even from other technology professionals. As technology experts and system power users, sysadmins are clearly not novice users; however, most software is designed with novices in mind [1]. Their broad areas of responsibility often result in a "juggling act" of sorts, quickly moving between tasks, and often not completing a given task from beginning to end in one sitting [2].

Also differentiating system administrators from regular end users of computer systems is the environment in which they work. As more business is conducted over the Internet, simple two-tier architectures have grown into complex n-tier architectures, involving numerous hardware and software components [3]. Because this infrastructure must be managed nearly flawlessly, the industry has seen system management costs exceed system component costs [4-6]. In addition, any system downtime can result in significant monetary losses. Although many vendors are exploring automated system management to cope with these complex and risky environments [4, 7, 8], these tools offer little comfort to system administrators, as the sysadmins are often held responsible for any system failures [6].

Citing the unique problems they face because of the complex systems they manage, their risky work environment, and their power-user access, authorities and skills, Barrett et al. [9] call for a focus on system administrators as unique users within HCI research. By examining the work practices of sysadmins, practitioners can design and develop tools suited to their specific needs. With the human cost of system administration now exceeding total system cost [4], the importance of catering to these specialized users is apparent.

This paper proceeds with a description of our research study and a narrative of system administrator work practices and work environment in Section 2. Section 3 presents the findings of our research study and the implications these findings have on tools designed for system administrators. Next, Section 4 presents a modified user satisfaction model that proposes a link between the tool implications that emerged from our research study – system attributes – and the impact of these attributes on system use. Section 5 concludes with a summary of the findings of this study.

2. WORK PRACTICES OF SYSTEM ADMINISTRATORS
This section describes our research study and explores the context in which system administration work is done and the work

CHIMIT'08, November 14–15, 2008, San Diego, California, U.S.A.

practices of Steve[1], a system administrator we shadowed. This narrative is augmented with responses from semi-structured interviews we conducted with sysadmins in both industry and academic organizations.

2.1. Research Study

Our research study was conducted over the past year and utilized a multi-method approach, including: on-site observation of a senior system administrator, semi-structured interviews, and a review of previous system administrator research. Our study participants included both junior and senior system administrators whose work responsibilities included the administration of networks, storage, operating systems, web hosting, and computer security. The system administrators we studied worked in enterprise or university settings. Our observations of and conversations with our participants allowed us to observe not only the work of system administration, but gain a better understanding of how the work is accomplished. Semi-structured interviews gave us the opportunity to ask more pointed questions about the sysadmins' motivations and reasons for their particular work routines and allowed us to collect their opinions on why they choose to use or not use a given tool to accomplish their work. With the insights we gained from these investigations, we turned our efforts to a review of the existing system administrator studies to confirm our findings.

2.1.1. Shadowing

To begin our investigation, a system administrator was shadowed for two days. Due to privacy and security concerns, no video, audio, photographs, or artifacts were collected. A personal laptop was not allowed in the server room for the researcher to use and would have proved cumbersome when following the subject; therefore, observations were limited to hand-written notes taken by the researcher. The researcher took extensive notes during the two shadowing sessions, which were expanded each evening. These notes were then coded for attributes and characteristics that were important to the work of the system administrator, whether they were present in the tools used or not. A working list of common themes was created and referenced during the coding of the interview questions and literature review.

2.1.2. Interviews

Semi-structured interviews were conducted to continue our investigation into tool characteristics important to system administrators. Six interviews were conducted with current system administrators at a location convenient to the sysadmin, two in person at the sysadmin's place of work and four over the phone. Three of the sysadmins interviewed worked for a Fortune 500 services computing company, while the other three worked in an academic setting, one for a large university and the other two for a college within the same university. The average length of time as a system administrator was 14 years (ranging from 8 years to 25 years) and the average age was 39 (ranging from 30 to 58 years old). Consistent with demographics reported by SAGE (91.6% male, 8.4% female), most of our participants (83%) were male [10]. The interviews were conducted at the convenience of the sysadmin and addressed they work they did, the tools they used, and reasons why they personally would use or not use a given tool in their jobs. Due to security and privacy concerns, interviews were not audio taped and recorded responses were limited to copious notes taken by the researcher. Interview notes were reviewed and expanded immediately following the interview to make sure all responses and relevant information were captured. These documents were then coded for characteristics important to and/or necessary to the system administrator in their work.

2.2. Environment

Steve is a senior operating systems administrator and works in a research division of a Fortune 500 company. He is responsible for maintaining an infrastructure that is comprised of multiple components (e.g., database management servers, application servers, and web servers) residing on hundreds of servers distributed across many networks and running on multiple operating systems. The primary network over which Steve shares responsibility is physically located at two separate sites, approximately 15 miles apart. Additional network responsibilities include seamless interfacing with the research networks at the organization's other worldwide research centers and the organization's global intranet. Disaster recovery and backup systems add complexity to this environment and the requirements for data availability have amplified both the importance of continuously available data and the cost of system downtime. Though many aspects of this system are automated, Steve and his colleagues are still ultimately responsible for the management and coordination of the entire system. Unexpected downtimes are not tolerated by the research staff and any loss of data could impact the research efforts of many different teams and is considered unacceptable.

We conducted interviews with sysadmins working for an international Fortune 500 Company that offered system administration outsourcing services. These interviewees were senior system administrators that led teams of four to six other sysadmins and were themselves involved in the more complex system administration tasks. These sysadmins had experience in many different aspects of system administration, but currently focused on operating system administration, web server administration, and large database administration. Similar to the Steve, the service computing sysadmins we interviewed reported working with similarly complex infrastructures and network configurations for the large companies they supported, but with more stringent requirements for data security and data availability. In their work, service level agreements (SLAs) assign specific and costly consequences to any system downtime outside of predetermined change windows. Any loss of data would cause significant customer impact and is considered unacceptable.

We also interviewed system administrators at a large university in the southwest United States. One person we intervived worked with the University's system administration department, while the others were employed by a college within the University. The University sysadmin was an operating system administrator and reported that the computing environment he supported was large and complex and supported by "dozens" of system administrators, similar to the environment reported by our Fortune 500

---

[1] To protect the anonymity of our participants, names and some personal details have been changed

interviewees. The College sysadmins reported broad responsibilities for the overall system, in contrast to the sysadmins in large organizations whose responsibilities were focused onto a function, operating system or application. The environment supported by these sysadmins was smaller in terms of servers and connections supported, but was maintained by only two to four sysadmins. An academic environment offered more flexibility in system downtime, but often introduced more infrastructure complexity because of the variety of services and products supported at different levels of university organization. For example, the College system administrators needed to support the computing infrastructure specified by the College while maintaining interoperability with the University computing infrastructure, which was specified by the University and often used different components and security restrictions.

2.3. Work Practices

With key responsibilities that include hardware, networking, and operating systems, the work of system administration consists of two major activities: system maintenance and system planning. System maintenance is a continual process that includes monitoring of system state and troubleshooting of any problems that arise. Monitoring activities are often automated by a number of tools that continually watch for changes in the system, such as component performance or disk utilization, and generate notifications when predetermined thresholds are reached. These notifications can appear on a server monitor or be emailed to a member of the team. Even with the availability of a monitoring function in some tools, other tools did not contain this functionality and Steve would check system status either manually (usually using the command line) or, many times, by scripting a tool to fill this need. When a problem with the system arose that Steve couldn't immediately solve, he would note it in Bugzilla, an open source web-based tool used for problem tracking, where it was then visible to his colleagues. Interviewees reported similar monitoring tools and activities, relying on this functionality to let them focus on the task at hand with some reassurance that "the rest of the system is being watched."

Troubleshooting work is an on-demand activity, initiated by system errors and often discovered by monitoring tools, though troubleshooting was also observed during the installation of new system components. This problem-solving work relies on the system administrator's knowledge of the problem and understanding of the overall system. Probable solutions are hypothesized based on information in system logs and dumps and previous experience, and often implemented in a trial-and-error fashion. After each potential solution was executed, Steve would issue a command to check the current system state and would occasionally check the logs for a verification of the action that was taken. When troubleshooting a problem, Steve commented that the best sysadmins had a good sense of their system and a keen intuition of probable solutions. Similar to responses from interviewees, troubleshooting efforts were augmented by fellow sysadmins (both inside and outside of the company), internal knowledge bases, and Google searches, though the usefulness of many internet search results were judged by the source of information.

System planning includes future system design and capacity planning, and system upgrades and installation. All research participants reported the work of system design and capacity planning as a responsibility of senior system administrators. As a senior system administrator, Steve participated in this activity and met with management and key users to determine future system needs. Once needs were defined and approved, Steve would research potential solutions and make recommendations to management.

System upgrades and installation activities were performed by all levels of sysadmins. This work included all activities related to the addition of new system components, that is, setting up hardware and wiring, installation, and configuration. Steve remarked that "something always breaks" with the addition of a new component because of the unknowns introduced into the system, and that similar unreliable behavior isn't tolerated in tools.

The work of system administration is collaborative in nature. All system administrators observed and interviewed reported belonging to a team of sysadmins, each with their own areas of expertise. System problems were often solved with other sysadmins, utilizing instant messaging, listservs, email, and phone calls. Regardless of their area of expertise, each sysadmin was expected to have a constant understanding of both general (e.g., current system state) and specific (e.g., the names and locations of "their" data backup servers) system information.

The workdays with Steve were started with a plan of tasks to accomplish, but these plans were quickly interrupted and changed. System errors with higher priority forced other tasks to wait and many tasks were started and left numerous times before completion. Our other research participants also noted that the work of system administration is one that is constantly marked with interruptions and multitasking. In one typical period, Steve was checking the status of an earlier system problem in Bugzilla, on the telephone with a hardware vendor discussing a new problem, and checking on the wiring of a server that was experiencing intermittent downtimes.

Steve used many tools in his work, and often commented on the high number of tools needed to do his job, many of which offered him limited information or functionality. As with our interview participants, he primarily used IM and email to collaborate with other sysadmins on his team, though he was issued a cell phone by his organization for emergencies. When analyzing system information, Steve would use the tool that uniquely suited his needs. For example, when checking on system utilization, he stated his preference for a GUI reporting tool because it presented the information quickly and in an easy-to-understand format. However, when checking other system statistics, his GUI (graphical user interface) tools were too slow to present the information and he preferred the flexibility of CLI (command line interface) commands. His team maintains system information in spreadsheets, in their own implementation of Bugzilla, and in organizational-supported knowledge bases.

Throughout our time spent with Steve, it was obvious that many of the tools available to him were not always practical in his work environment. CLIs were preferred to check system state because of their guarantee of current system information, while GUIs were used when complicated – yet system defined – graphical reports were needed. When asked if he preferred CLIs to GUIs, Steve answered that he would use whatever tool would get him what he needed the fastest, whether it be a report (in this case, a GUI tool) or the execution of a system task (in another example, a custom CLI script).

Interview responses echoed similar opinions of the tools available. Some participants thought the amount of tools available to them was "about right," while others thought they needed too many tools to do their jobs. While many stated a preference for CLI tools, this was explained as an underlying need for tools that were fast (both in speed of execution and speed of information gathering), used little system memory, and were scriptable. Great importance was placed on reliability of tools, with one sysadmin noting, "if it doesn't work once, it's out," and he'll find another tool to do that particular task.

3. IMPLICATIONS FOR TOOL DESIGN

It is clear from the description presented above that the work environment and work practices of system administrators both impact and are impacted by the tools they use. Furthermore, because of the technical nature of their work, it is impractical to study system administrators without also studying the tools they must use to do their jobs. Suchman [11] makes the argument that in order for computer systems to be practically used in context, their design must be in line with the underlying work practices of those who use them.

To augment our own observations and findings, a review of previous system administrator studies was conducted. Our review process is described below, followed by a discussion of our findings and the implications that these finding have on tools designed for system administrators.

3.1. Review of System Administrator Studies

To complete our analysis, a review of the relevant literature was conducted. Selected publications were limited to journals and conference proceedings, excluding articles in trade magazines. Studies included were those that met the following criteria:

1. The subjects of the study were exclusively system administrators, including web administrators, network administrators, security administrators, etc. Studies of system administration concepts with end users or students as subjects were excluded.
2. The study focused on the work, work practices, and/or tools of system administrators. Studies that focused on other issues, such as algorithms or computing environments, were excluded.

Because research involving system administrators is relatively recent, no date range limitations were used and studies were added to the sample iteratively. The literature reviewed was selected from CHIMIT 07 and an ACM search on the keywords "system administrator" and "system administration." Papers from the first CHIMIT (Computer Human Interaction for the Management of Technology) conference (CHIMIT07) were selected because of the conference's emphasis on system administrators and ACM publications were selected because of their known existing system administrator literature content. A forward search of these publications was then conducted using Google Scholar (scholar.google.com) to expand our search to include studies published outside of ACM.

Twenty-nine studies were identified in the search for relevant literature.[2] Each study was coded for characteristics important to system administrators. Because the goal of this classification was to identify system characteristics useful to system administrators, a characteristic was attributed to the study if 1) the characteristic was identified as an existing, useful aspect of the system(s) being used, or 2) the characteristic was identified as a missing aspect of the system, with the assumption that the missing characteristic was specifically identified by the authors because of an observed need for that characteristic.

3.2. Findings

The technical expertise of system administrators is apparent in narratives of their work and in our review of previous studies. As skilled workers, they have unique information and system needs from the tools they use. In her examination of computer applications in use, Bodker [1] states that the design of systems for experts is especially challenging because of the occupational knowledge required of the designers, and concluded that "what is a good user interface for those with a certain degree of competence may not be efficient for those with different levels of competence" (p. 172). Here, we summarize the findings of the research study and literature review.

The description of a system administrator's work environment shows that it is one that is complex, risky, and large scale. The sheer number of hardware and software components and their interactions in large systems are illustrative of the complexity of the environment. This work is also risky, because any system downtime or loss of data can result in significant organizational monetary loss or a personal loss of employment. Studies of systems used in airplane cockpits [12] and ethnographies of system administrators [e.g., 13] illustrate the need for *flexible* systems that are capable of as much variety as their systems support. Flexible systems are those that can adjust to new demands or conditions. As an example, a study of IT security administrators cited the need for flexible reporting [14], where users can specify sets or subsets of data to include and modify the output format to suit their changing environment and changing needs.

The large size of today's computing infrastructures presents a practical need for tools that can *scale* to meet the requirements of systems that continually grow and change. This need for tools that scale has been identified in other studies [e.g., 2, 15] and was named by system administrators themselves as a limitation of current tools [16]. An example of this need was depicted in a commercial for a storage and services company. Small business owners were shown launching their first website. These owners shared congratulations as their Internet traffic and online sales increased. However, this excitement quickly turned to concern and then panic once the product was featured on television and

[2] Due to article length limitations, it is impractical to cite all publications included in the review; however, a list can be requested from the first author.

their infrastructure crashed under the load of surging Internet traffic.

System *monitoring* activities were seen during system maintenance, with scripts, flashing screens, and email notifications used to alert the system administrator when certain predefined events occurred or thresholds were reached. Without monitoring capabilities, sysadmins must check various aspects of system and subsystem state at regular intervals as part of their system maintenance responsibilities. A common example of a monitoring function can be seen with automated emails that are sent when server capacity reaches a predetermined level. This email notification allows the sysadmin to focus on other aspects of the computing infrastructure he is supporting until the server capacity requires his attention.

As keepers of massive infrastructures, it is perhaps impossible for sysadmins to have a complete understanding of every component in their system [17]. However, a mental picture of the overall state of the system, or *situation awareness*, has been identified as a requirement in the work of system administration [3] and a survey conducted by Hrebec and Stiber [17] found that system administrators report a poor understanding of the computing infrastructures they support. Systems that can convey an understanding of overall system state while providing the ability to also access system details is important in the complex and dynamic infrastructures supported by system administrators. For example, when installing an operating system patch on a server, the sysadmin must know IP address and port information for the server (details) while also knowing where the server fits in the rest of the system and the overall system state.

A related system requirement can be found in the need for *accurate* and *current* information. An accurate understanding of one's system relies on accurate information, and many sysadmins reported relying on CLI commands because they reported information stored and calculated by the operating system and known to be accurate. For example, some sysadmins reported using tools in the past that reported incorrect I/O (input/output throughput) information, which was discovered only after noticing discrepancies and then confirming with detailed and time-consuming research. Current information is also a requirement in system administration, a need that was communicated by the system administrators we studied. For example, when asked about confirming the status of tasks, one participant reported completing a task using a GUI tool and then confirming the status using a CLI command. When asked why he would use a CLI command, he expressed concern that the GUI information might be cached but that the use of a CLI command would ensure current status information.

Many study participants reported the use of custom tools to monitor system state, configure components, and perform many other maintenance tasks, suggesting the need for *scriptability*. The ability to program, or script, portions of system administration work was apparent in two general situations: 1), when the automation of oft-repeated or complex functions generally executed through the command line, and 2), when the sysadmin wished for an api (a scriptable interface to a program) to automate some GUI tasks that took too long and required navigating multiple screens. The ability to script work processes has also been identified as an important attribute in previous studies [e.g., 15].

Troubleshooting activities illustrate the use of *logging* information while gathering information about the problem and *verifying* the outcomes of commands issued. The use of system logs was seen in the work of system administrators, particularly when executing long sequences of CLI commands or when returning to a task that was started, left, and then returned to later. In these cases, system logs were helpful to determine the last command issued or, in the case of verification information, to determine the outcome of the last command issued. The latter information was often used following installations or configurations as a way to verify action completion and assure the sysadmin that the task was complete. The need for additional information in complex and risky work is also supported in the literature. A study of government workers and their information seeking behavior done by Bystrom and Jarvelin suggests that as tasks increase in complexity, users are more likely to seek additional information [18]. Marketing research suggests that consumers use the search for additional information as a risk-reducing strategy [19]. The complex and risky work environment of system administrators suggests a greater need for information, seen the need for logging information and verification information.

One interviewee reported the use of a local installation of Bugzilla because it was an easily *accessible* central repository of system information. Originally created by Mozilla, Bugzilla is a free software application that is traditionally used to support software development by managing software bugs. The system administrator interviewed used his group's installation of Bugzilla to track system errors, outages, and component status. He reported his preference for Bugzilla over the corporate-sponsored knowledge base because of the ease with which he could access and edit information.

A study by Takayama et al. [20] found that *credibility* was an underlying factor in user interface choice. Based on comments from study participants, this research suggests that credibility may also be a factor in tool choice, regardless of user interface type. For example, many system administrators reported their preference for a given tool because they had used it for a long time or because himself or a respected peer had programmed the tool. When asked to elaborate on this motivation for tool use, many sysadmins identified the length of use and knowledge of the programmer as an indication of their confidence and trust in these tools.

The large number of tools used to gather small collections of information suggests that *data integration* could be useful. For example, a sysadmin troubleshooting a system error was seen accessing data from server logs, monitor notifications, vendor documentation, online forums, and Google searches. At one point, the sysadmin remarked that it "would be handy" to be able to access more than one piece of information from a single screen or application.

Additional characteristics that emerge in the discussion of tools used include *complete* information that is presented in a *format* that is easy to understand. One clear example of these two needs was seen when a sysadmin was trying to gain a better understanding of system usage. The sysadmin issued five CLI commands, stating he wanted to see "everything" on the screen and his GUI tool didn't report every metric he needed to access. After seeing the output from these commands, he switched to his GUI tool (that he had just avoided because of its lack of complete information) for its formatted reports and a graphical representation of the data he had just viewed. After asking why he used both tools, he responded that the CLI command gave him the detailed information he needed and the GUI tool provided him with a better overall picture of the system through formatted reports and graphical representations.

Given the hectic, multi-tasked nature of the work involved, it wasn't surprising to hear many of the participants report a preference for "whichever tool will get me what I need fastest." In fact, many system administrators expressed a preference for CLI tools simply because of their speed of tool start up and command execution. One sysadmin remarked that the only problems with his favorite GUI tool were the long start up time (which prompted him to rarely exit the program) and the system "hang" or delay that occurred when he was performing complex tasks.

It should be noted that while collaboration is an important part of system administration and its potential importance as a tool characteristic has been identified in previous sysadmin studies [2, 13], our research participants noted that they preferred it to be done using separate tools. In Steve's case, this may be due to the use of an internal organizational instant messaging client, but when asked about collaborative support in other tools, the system administrator expressed frustration at the effort required to manage and coordinate user ids, profiles, and especially contact information across multiple applications. He stated a preference to "stick to one [collaborative tool] and leave the rest alone." Additionally, one interviewee pointed out that integrated collaborative components oftentimes "slow down [his] already slow app" and he preferred lightweight external IM clients, such as IRC.

3.3. Important Characteristics

The strength of this focused investigation of technology-in-use lies in its ability to identify realistic solutions and guide potential designs [21]. By examining the work of system administrators, we have generated the following list of attributes that appear to be important to system administrators. (The reader should note that many attribute definitions were refined throughout the project, referencing the attribute definitions provided in [22].)

1. Flexibility: the way the system adapts to changing demands of the system administrator
2. Scalability: the ability of a system to scale to large and/or complex computing environments
3. Monitoring: the ability to monitor for certain events or conditions
4. Situation Awareness: the ability of a system to provide information about the overall state of the system
5. Scriptability: the ability to script add-ons or automate tasks provided by the system.
6. Logging Information: information that echoes or repeats previous actions taken
7. Accessibility: the ease with which information can be accessed or extracted from the system
8. Accuracy: the user's perception that the information is correct
9. Integration: the way the system allows data and functions to be integrated from various sources
10. Information Completeness: the degree to which the system provides all necessary information
11. Information Format: the user's perception of how well the information is presented
12. Information Currency: the user's perception of the degree to which the information is up to date
13. Speed: the degree to which the system offers timely responses to requests for information or action, including the speed of tool start up/initiation.
14. Reliability: dependability of system operation

Upon further inspection, these characteristics seem to fall into categories of attributes pertaining to attributes of the information supplied by the system and attributes of the system itself. This classification of characteristics can be seen in Table 1.

Table 1. Info and system attributes important to sysadmins

Information Attributes	System Attributes
Logging	Flexibility
Accuracy	Scalability
Completeness	Monitoring
Format	Situation Awareness
Currency	Scriptability
	Accessibility
	Integration
	Speed
	Reliability

4. MODEL AND THEORY

Although the above list of characteristics important to system administrators is interesting, it does little more than summarize observations and offer untested guidance to designers. Without evidence that these characteristics will influence a system administrator to use a particular tool, practitioners will be reluctant to invest the time and money needed to implement these features. The goal of this study is to understand the link between these characteristics and their impact on system administrator perceptions – and ultimately, use – of the system.

When evaluating user perceptions of information technology [e.g., 23] theories of technology acceptance [e.g., 24, 25] and user satisfaction [e.g., 26, 27] are often cited, and each has its strengths and limitations. The strength of technology acceptance theory lies in its ability to link behavioral attitudes and beliefs about the system (i.e., ease of use and usefulness) to users' system usage behaviors. That is, technology acceptance theories have high predictive ability. A limitation of technology acceptance theories, however, is the lack of system feedback that can be easily translated into design guidelines. For example, feedback regarding a system's ease of use and usefulness offers no concrete information about what attributes and features to include in system design. However, these theories offer little practical guidance on system design. The opposite is true of user satisfaction theories. The strength of user satisfaction theories lie in the measured impact of system attributes, such as information format and system accessibility, on user satisfaction. That is, system attributes that are found to be significant indicators of user

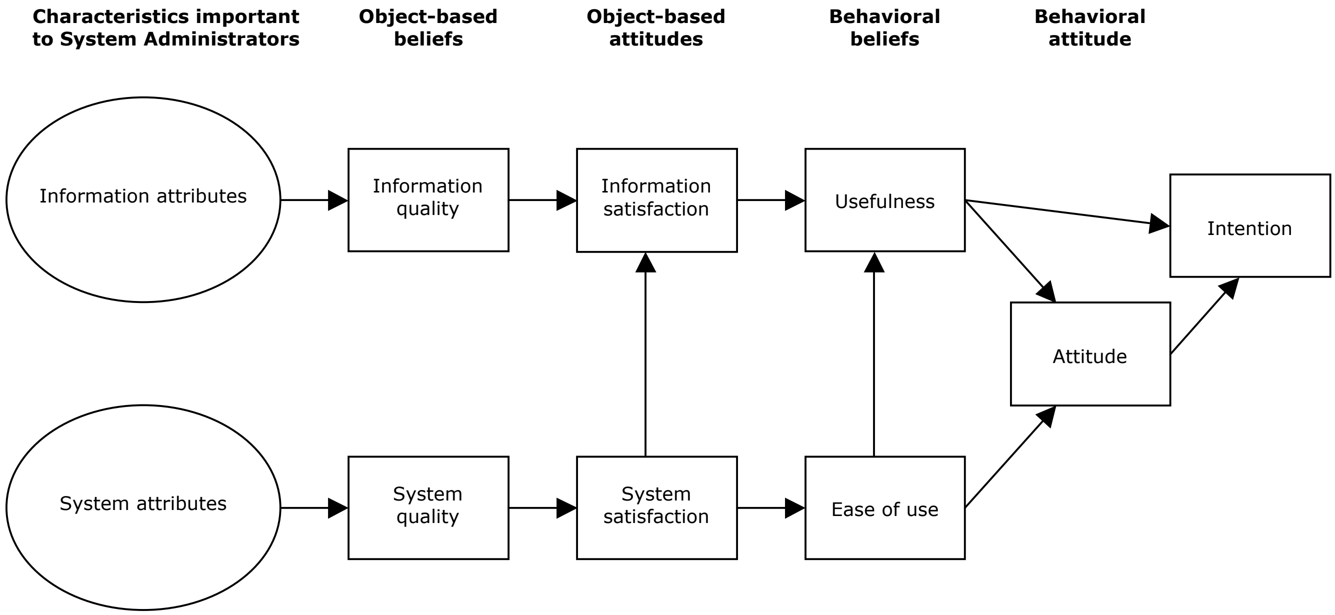

Figure 1. Integrated user satisfaction model

satisfaction can be addressed in the design of the system. The limitation of user satisfaction theories is seen in their poor predictive ability. That is, a user's satisfaction of a system has been shown to be a poor predictor of their actual system usage behaviors [e.g., 22, 24].

To take advantage of the strength of both the technology acceptance and user satisfaction theories, Wixom and Todd [22] presented a modification of DeLone and McLean's original user satisfaction model [23] that links system and information satisfaction with the behavioral predictors found in technology acceptance literature, perceived ease of use and usefulness. They argue that the object-based attitudes and beliefs expressed in system quality, information quality, system satisfaction, and information satisfaction affect the behavioral beliefs that are captured in ease of use and usefulness. These behavioral beliefs, in turn, influence a user's behavior (i.e., their use or non-use of a system). Essentially, this new model represents a theoretical integration of user satisfaction and technology acceptance theories. The strength of the model lies in its ability to guide IT design and development and predict system usage behaviors. System and information quality antecedents offer concrete attributes important to the user that can be addressed and tested throughout the system development lifecycle. (See Figure 1.)

This model also provides a solid foundation for investigations into software characteristics that might be more or less important to specific user groups, such as system administrators. Because system administrators are still computer users in the general sense, we expect the overall theoretical model to hold. Their unique work environment, technical background and job requirements, however, suggest that they may have different needs when using computers or software applications to do their jobs. Previous studies (e.g., [26, 28-30]) have focused on a relatively small number of characteristics that, although telling in their underlying structure [22], have been criticized for investigating arbitrary system attributes [31]. The analysis of system administrator work practices above identifies system and information quality attributes (i.e., antecedents) that are meaningful and important to system administrators.

Wixom and Todd [22] identify information and system quality antecedents through a decomposition and integration of factors used in prior user satisfaction studies. The authors suggest that while the antecedents used in their study are generally applicable, the importance of each may be dependent upon the specific settings and systems studied. We agree with the statement, and argue that the unique environment, work practices, and technical ability of system administrators may introduce additional attributes. In fact, many of the system and information quality attributes uncovered in our examination of system administrators and their work practices have been identified by Wixom and Todd [22] (see Table 2).

A few differences in repeated attributes (identified in Table 2 with an asterisk) should be discussed. First, Wixom and Todd define integration as "the way the system allows data to be integrated from various sources," while we have expanded the definition to include the integration of functionality. Additionally, Wixom and Todd's study of data warehouse users found timeliness not be a significant indicator of system quality. These findings are supported by work done by Bodker [1], which found that the speed of an information system was not important to novices, but was important to skilled workers.

Table 2. Information and system attributes

	Information Attributes	System Attributes
Wixom & Todd	Completeness Accuracy Format Currency	Reliability Flexibility Integration* Accessibility Timeliness (Speed)*
Additional Attributes	Log Information	Scriptability Scalability Situation Awareness Monitoring

The remaining additional attributes are worthy of discussion. The need and use of logging information in the work of system administration is apparent in observations of their work. A study done by Bystrom and Jarvelin [18] suggests that as tasks increase in complexity, users are more likely to seek additional information. Marketing research suggests that consumers use the search for additional information as a risk-reducing strategy [19]. The complex and risky work environment of system administrators suggests a greater need for information, seen the need for logging information.

The ability to program, or script, portions of system administration work was apparent in two general situations: 1), when the automation of oft-repeated or complex functions generally executed through the command line, and 2), when the sysadmin wished for an api (a scriptable interface to a program) to automate some GUI tasks that took too long and required navigating multiple screens. The ability to script work processes has been identified as an important attribute by Haber and Bailey [15].

The sheer size of today's computing infrastructures presents a practical need for systems that can scale. This system requirement has been identified in ethnographies of system administrators [e.g., 2, 15] and has been identified as a limitation of current tools [16].

As keepers of massive infrastructures, it is perhaps impossible for sysadmins to have a complete understanding of every component in their system [17]. However, a mental picture of the overall state of the system has been identified as a constant requirement in the work of system administration [3].

Finally, monitoring capabilities allow system administrators to be alerted when pre-defined events occur or thresholds are reached in various components or subsystems without having to mentally process massive amounts of information available in logs and system dumps. This ability to define rules that perform tasks (e.g., send an email) to reduce information overload has been shown to be useful to both novices and expert users [32].

In summary, the unique work and context of system administration suggests some information and system attributes presently missing from user satisfaction research. The model presented in this paper suggests that design characteristics can be linked to system usage behaviors and identifies attributes important to system administrators.

The proposed model will be tested using a survey methodology and will target system administrators of all experience levels and specialization areas. Where possible, measures will be adopted from previous research. Construct items for new measures will be developed following the methodology described by Churchill [33]. The survey will be posted on professional system administrator messageboards and data will be collected through online surveys. The data will be analyzed using structural equation modeling, given that this method allows us to test complex causal models.

5. CONCLUSION

This study presents the work practices of system administrators. As skilled workers and power users responsible for large computing infrastructures, their work is fast-paced and is characterized by multitasking, task juggling, and collaboration. The complex and risky work environment in which they operate distinguishes them from regular computer users. System administrators require absolute attention to some details while maintaining a general understanding of the overall system. For example, when installing an operating system patch on a server, the sysadmin must know IP address and port information for the server (details) while also knowing where the server fits in the rest of the system and overall system state. These factors indicate that system administrators may be unique users with information system requirements that are different from the requirements of regular computer users.

Our analysis of system administrator work practices and environment has implications for tool design. The study identified system and information quality attributes that are new and differentiate this unique user group. Log information, monitoring, situation awareness, scalability, and scriptability have not been previously identified as important tool attributes and intuitively seem important primarily to system administrators. For example, regular computer users rarely require scripting abilities in the systems they use or rely on system logs, while system administrators require support for these activities in their work.

Finally, we presented a modified user satisfaction model that links system design attributes to end user satisfaction and system use, presenting an opportunity to measure the impact that these identified attributes have on system administrator beliefs and tool usage. We believe that this study provides researchers guidance for adapting existing user information satisfaction models for tools used by system administrators. Work is under way to empirically test the proposed model to assess the relationships in the context of system administration.

6. REFERENCES

[1] Bodker, S. 1989. A Human Activity Approach to User Interfaces. Human-Computer Interaction, 4(Jan 1 1989), 171-195.

[2] Barrett, R., Kandogan, E., Maglio, P., Haber, E., Takayama, L. and Prabaker, M. Year. Field Studies of Computer System Administrators: Analysis of System Management Tools and

Practices. In Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW '04) (Chicago, IL, USA, Nov 6-10 2004, Year). 388-395.

[3] Bailey, J., Etgen, M. and Freeman, K. 2003. Situation Awareness and System Administration (Workshop). In Proceedings of the Conference on Human Factors in Computing Systems (CHI' 03) (Ft Lauderdale, FL, USA, Apr 2003, 2003).

[4] IBM. Autonomic computing: IBM's perspective on the state of information technology. http://www.ibm.com/industries/government/doc/content/resource/thought/278606109.html Accessed Jan 2006.

[5] Kephart, J. and Chess, D. 2003. The vision of autonomic computing. IEEE Computer, 36, 1 2003), 41-51.

[6] Patterson, D., Brown, A., Broadwell, P., Candea, G., Chen, M., Cutler, J., Enriquez, P., Fox, A., Kiciman, E., Merzbacker, M., Oppenheimer, D., Sastry, N., Tetzlaff, W., Traupman, J. and Treuhalf, N. Year. Recovery-oriented computing (ROC): Motivation, definition, techniques, and case studies. In Proceedings of the Technical report CSD-02-1175 (Year).

[7] HP. Adaptive Infrastructure. http://h71028.www7.hp.com/enterprise/cache/5003-0-0-0-121.aspx. Accessed Jan 2006.

[8] SunMicrosystems. N1 Grid System. http://wwws.sun.com/software/solutions/n1. Accessed Jan 2006.

[9] Barrett, R., Chen, Y. Y. M. and Maglio, P. 2003. System Administrators are Users, Too: Designing Workspaces for Managing Internet-Scale Systems (Workshop). In Proceedings of the Conference on Human Factors in Computing Systems (CHI' 03) (Ft. Lauderdale, FL, USA, Apr 2003, 2003).

[10] SAGE. 2006. SAGE Annual Salary Survey 2005-2006. 2006.

[11] Suchman, L. A. 1995. Making Work Visible. Communications of the ACM, 38, 9 (Jan 1 1995), 56-64.

[12] Dalcher, D. Year. Trust, Systems and Accidents: Designing Complex Systems. In Proceedings of the International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS '03) (Huntsville, AL, USA, Apr 7-10, Year).

[13] Haber, E. and Kandogan, E. 2007. Security Administrators in the Wild: Ethnographic Studies of Security Administrators. SIG CHI 2007 Workshop on Security User Studies: Methodologies and Best Practices2007).

[14] Botta, D., Werlinger, R., Gagné, A., Beznosov, K., Iverson, L., Fels, S. and Fisher, B. Year. Towards Understanding IT Security Professionals and Their Tools. In Proceedings of the Symposium on Usable Privacy and Security (SOUPS '07) (Pittsburgh, PA, USA, Jul 18-20, Year). 100-111.

[15] Haber, E. and Bailey, J. Year. Design Guidelines for System Administration Tools Developed Through Ethnographic Field Studies. In Proceedings of the Computer-Human Interaction for the Management of Information Technology (CHIMIT '07) (Cambridge, MA, USA, Mar 30-31, 2007, Year).

[16] Verdoes, J. A. Year. In Search of a Complete and Scalable Systems Administration Suite. In Proceedings of the Large Scale System Administration of Windows NT (Seattle, WA, USA, Year). USENIX Association.

[17] Hrebec, D. G. and Stiber, M. Year. A Survey of System Administrator Mental Models and Situation Awareness. In Proceedings of the ACM Conference on Computer Personnel Research (SIGCPR '01) (San Diego, CA, USA, Jan 1, Year). 166-172.

[18] Bystrom, K. and Jarvelin, K. 1995. Task complexity affects information seeking and use. Information Processing and Management, 31, 2 1995), 191-213.

[19] Murray, K. B. and Schlacter, J. L. 1990. The Impact of Services Versus Goods on Consumers' Assessment of Perceived Risk and Variability. Journal of the Academy of Marketing Science, 18, 1 (Jan 1 1990), 51-65.

[20] Takayama, L. and Kandogan, E. Year. Trust as an Underlying Factor of System Administrator Interface Choice. In Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '06) (Montreal, Quebec, CA, Apr 22-27, 2006, Year). 1391-1396.

[21] Button, G. and Harper, R. 1996. The Relevance of 'Work-Practice' for Design. Computer Supported Cooperative Work (CSCW), 4(Jan 1 1996), 263-280.

[22] Wixom, B. H. and Todd, P. A. 2005. A Theoretical Integration of User Satisfaction and Technology Acceptance. Information Systems Research, 16, 1 (Mar 2005 2005), 85-102.

[23] DeLone, W. and McLean, E. 1992. Information Systems Success: The Quest for the Dependent Variable. Information Systems Research, 3, 1 (Mar 1992 1992), 60-95.

[24] Davis, F. D. 1989. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. Management Information Systems Quarterly, 13, 3 (Sep 1983 1989), 319-340.

[25] Venkatesh, V. 2000. Determinants of Perceived Ease of Use: Integrating Control, Intrinsic Motivation, and Emotion into the Technology Acceptance Model. Information Systems Research, 11, 4 (Dec 1 2000), 342-365.

[26] Bailey, J. and Pearson, S. W. 1983. Development of a Tool for Measuring and Analyzing Computer User Satisfaction. Management Science, 29, 5 (May 1983), 530-545.

[27] Shneiderman, B. 1997. Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison Wesley, Redwood City, CA.

[28] Baroudi, J. and Orlikowski, W. 1988. A short-form measure of user information satisfaction: A psychometric evaluation and notes on use. Journal of Management Information Systems, 4, 4 1988), 44-59.

[29] Doll, W. and Torkzadeh, G. 1988. The measurement of end-user computing satisfaction. Management Information Systems Quarterly, 12, 2 (Jun 1 1988), 259-274.

[30] Ives, B., Olson, M. H. and Baroudi, J. 1983. The measurement of user information satisfaction. Communications of the ACM, 26, 10 1983), 785-793.

[31] Galletta, D. and Lederer, A. L. 1989. Some cautions on the measuremnet of user information satisfaction. Decision Sciences, 20, 3 1989), 419-439.

[32] Maes, P. 1994. Agents that reduce work and information overload. Communications of the ACM, 37, 7 1994), 30-40.

[33] Churchill, G. A. 1979. A Paradigm for Developing Better Measures of Marketing Constructs. Journal of Marketing Research, 16, 1 (Feb 1979 1979), 64-73.