PJIM PARSONS JOURNAL FOR INFORMATION MAPPING

THE PARSONS INSTITUTE
FOR INFORMATION MAPPING

68 Fifth Avenue
New York, NY 10011

212 229 6825
piim.newschool.edu

Measuring the Value of Big Data Exploitation Systems: Quantitative, Non-Subjective Metrics with the User as a Key Component

JANA SCHWARTZ, PHD
JOSHUA C. POORE, PHD
EMILY VINCENT, MS
DAVID REED, MS

KEYWORDS *Big data analysis tools, metrics, cognitive load, task performance, applied psychophysiology, engagement, user experience, user-centered*

DATE *Fall 2012–Present*

URL *FOSS software available from*
https://github.com/draperlab/xdatalogger

ABSTRACT A big data visualization system is comprised of a diverse set of components. Obvious components include the data set, an analytic acting on the data set, and the visualization of the data set and/or results from the analytic, with one or more of each of these components in the system. The consumer, or user, of the visualization is a less obvious component of this system, but is nonetheless critical to understand in gauging system performance.

There are well-defined, quantitative metrics to describe the data set (how big?, what kind of data?), the analytic (how quickly does it converge?), the visualization (how over-plotted?), and the IT infrastructure (how much memory?). However, the user operates within an additional set of constraints provided by her context —the performance of the supporting IT infrastructure, time constraints, distractions, etc. As such, understanding the user-in-context has typically employed subjective and qualitative approaches (did you enjoy the presentation?) rather than objective and quantitative as seen in the other measures of the system. Such approaches result in inappropriate holistic assessments of systems with some components measured quantitatively (and often in great detail) while others are measured qualitatively, altogether resulting in unforeseen potential risks in overall system performance.

Draper Laboratory is developing quantitative, non-subjective measures to describe system-level performance within and between composed big data systems. These measures leverage classical psychometrics, but extend to include dynamic behaviors and physiologic responses. We have previously developed a measure of Engagement, and are currently working to apply that, and related measures, to assess the performance of big data analysis systems.

INTRODUCTION

Big data exploitation systems (BDES) are composed of a common set of components: one or more data sets; the analytics applied to the data; the visualizations by which the data and processed information are presented to the user; and the infrastructure upon which all of these computations are run. The user is also a component of a BDES—an innovative component, who enables the other aspects of the system to produce value. A BDES operates within the construct of a mission: there is some goal to be achieved, which may range from data mining

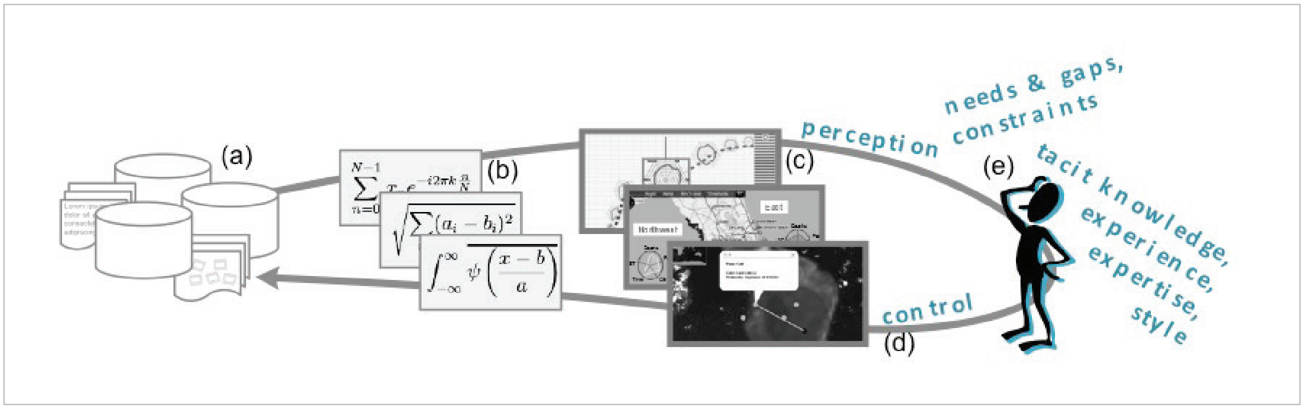


FIGURE 1: *The components of a Big Data Analytics Systems:* (a) *data set;* (b) *analytics;* (c) *visualizations;* (d) *infrastructure;* (e) *user*

BDES COMPONENT	(A) DATA SET	(B) ANALYTIC	(C) VISUALIZATION	(D) INFRATRUCTURE	(E) USER
COMPONENT-LEVEL DESCRIPTIVE METRIC	Feature basis (*e.g., time, location, content, metadata*)	Appropriateness of data sets (*time series, transactional, database*) Type of output (*feature, graph, probability*) Order of computations required (*e.g., O(n)*)	Total bases spanned Bases spanned continuously	HIDDEN: • # computing nodes • Processor speed • Total memory • Communications bandwidth • Up-time UI: • Screen size • Input methods	IQ Years of experience Personality Cognitive style Executive Function Motivation
COMPONENT-LEVEL PERFORMANCE METRIC	Volume Velocity Variety Veracity	Time to convergence Iterations to convergence Confidence interval Final error	Number of points drawn Color map scaling Level of overplot Precision/correctness	HIDDEN: • Performance vs. benchmarks UI: • Human factors metrics • Heuristic evaluation • Cognitive walkthrough scores	Reaction time Accuracy Affect Cognitive Load Attention Engagement Physiology

TABLE 1: *Summary of Descriptive and Performance Component-level Metrics for BDES Components*

to hypothesis testing. Each of the components, and the mission context, may be defined and developed independently. However, performance of the BDES must be measured at the system level.

In this paper, we introduce the components and context of a BDES. We introduce descriptive and performance metrics for each. We then motivate why limiting one's thinking to this component level is insufficient and potentially detrimental– the complexities of big data provide incredible possibility to solve previously unanswerable questions, but that possibility is coupled with risk if technologies and data are misapplied. We then propose a solution for achieving quantitative, non-subjective, system-level measurement with sufficient resolution and scope to enable meaningful performance assessment, by including the user as a key component.

THE COMPONENTS AND CONTEXT OF BIG DATA EXPLOITATION SYSTEMS

Every big data exploitation system (BDES) is composed of the same set of core components (see FIGURE 1). In this section, we describe each component, as well as the mission context. We summarize the descriptive metrics (typically considered during design and integration activities) and the performance metrics (relevant to operational use) in TABLE 1.

THE DATA

First, a BDES requires one or more data sets to operate upon (FIGURE 1, far left, labeled (A)). Classically, the most common data sources to exploit were of time-series data: one or more lists of values, as concurrent or coincident functions of time. These are the data sets of physics, and may be as small as a student's lab notebook or as vast as the data from the Large Hadron Collider (presently ~25 Petabytes per year, *http://wlcg.web.cern.ch/*). Time-series vectors are arguably the simplest form of structured data—two (or more) pieces of continuous numeric data in each record describing time and according value—and classical statistics approaches were developed with such data in mind. However, data need not be continuous nor numeric: structured data, typically organized in a table or database, may further contain a mix of continuous and categorical, cardinal, ordinal, and nominal data, organized in a consistent manner. For example, a set of billing data for a new small business may include just a few records, containing date/time information along with names, addresses, products, and prices. Small amounts of structured data may be easily manageable on a desktop computer; for perspective, Google developed the MapReduce algorithm to enable management of a database on a scale commensurate with the Internet.[1]

Unstructured data sets—including free text data (from the short text of social media to the prose of professional reporting), digital imagery, audio, and video data—present an additional level of challenge, and are typically pre-processed with the use of domain-specific algorithms. For instance, text data may be translated or transliterated across languages. It may be searched for content (e.g., keywords), resulting in frequency distributions of count data for content category instances, or just raw counts of interesting words. It may be also assessed for sentiment and polarity, producing numeric scores[2]—thus transforming an unstructured, complex data type into structured data. Digital images can similarly be searched for content (e.g., faces) or aesthetic quality (e.g., color, tone, texture), albeit with more complex algorithms. The same pattern applies to audio data (such as a song or podcast), which can be searched for speech content of interest, language, or basic tonality.[3] Video data is a complex blending of audio and imagery; it can be additionally processed to correlate imagery and audio features, and to track moving objects within a scene. Overall, few sources or types of data are intractable from an analytic standpoint; however, "big data" has never really been about the type of data one has.

The challenge of processing of large amounts of data is not new—supercomputers have been in use for this sole purpose since the 1960s, and continue to evolve. Accordingly, "big data" should not be perceived as a static concept, instead it evolves to reference current data challenges. Fifty years ago, entire rooms of supercomputers had orders of magnitude less storage and processing than the smart phones we carry in our pockets today. A team of researchers from IBM perhaps best captures the essence of today's big data problem with their mnemonic "The Four v's of Big Data."[4] The v's stand for Volume ("how much data?"), Velocity ("how quickly is it being generated and/or exploited?"), Variety ("how varied is the data across or within records?"), and Veracity ("how directly does the data relate to the problem we are trying to solve, or how much do I trust a particular source?"). These four v's describe how a data set is measured within a BDES. High velocity and large volume data may stress the capabilities of the infrastructure to ingest and store the data, respectively. High variety and veracity data sets may require innovation in the development of and integration with analytic and visualization components.

ANALYTIC(S)

Moving to the right in the FIGURE 1 (B), the next component of a BDES is the analytic, or set of analytics, used to transform the raw data into a more useful form. Coverage of the space of possible analytics is well beyond the scope of this paper: analytics presently applied to big data sets range from classical statistical methods to graph analytics; machine learning techniques enable classification of each record with respect to the corpus, while inferential techniques impute missing data.[5] An analytic may run in real time (e.g., on high velocity streaming data, and/or under direct user manipulation) or in an offline mode, asynchronously with the user-in-the-loop process. It may be described in terms of its appropriateness to the data type, class of output, or computational scalability. Performance of an analytic may be measured in time to convergence, error at convergence, or as a trade between accuracy and error such as a receiver operating characteristic (ROC) curve.

VISUALIZATION(S)

The third component (C) in FIGURE 1 depicts the visualization or set of visualizations in the BDES. The visualization is often the user's only window into the raw data and the output of the analytics. As with analytics, there are many different ways of taxonomizing visualizations, which are well beyond the scope of this paper.[6] Visualization of numeric data, at moderate Volume and Velocity, is generally well-understood: a timeline is a one-dimensional view; a scatter plot can provide two or three dimensions along its axes, with embedded dimensions as data glyphs; parallel coordinate views easily show many dimensions; tree views can show multiple dimensions of hierarchy. Many qualitative or categorical data features can be elegantly transformed into numeric bases to enable familiar visualization. New methods for compact visualization of data with ranging Variety and Veracity are constantly being developed in order to present correlations across bases more effectively (see, for example, *http://www.d3js.org* for a sampling of open source web-based components). As an example, a word cloud is histogram represented in a nontraditional basis, rather than linear space (a bar chart) or radial coordinates (a pie chart). It is then the role of the user to derive meaning from the correlations these views highlight, and make a case for causality (or not). Visualizations may be standalone or may cross-filter with other views across feature bases. Visualizations may be measured in many ways, including the bases spanned continuously, number of points drawn, level of overplot, and precision/correctness.

INFRASTRUCTURE

Latent in FIGURE 1 (D, the arrow) is the infrastructure upon which the BDES runs. The infrastructure must host or access the data, support the processing of the analytics, and support the rendering of the visualizations. The infrastructure additionally provides the physical interface between the virtual components of the BDES (as described in the previous three sections) and the user (discussed next). This interface informs how the user will physically interact with the rest of the BDES. It informs and constrains the users' behaviors, and the way they make sense both the interface and the data. For example, a large multi-touch screen interface provides a very different method for triage of a database (an idea under development by the USC ICT Mixed Reality Lab, *http://mxrlab.com*) than a text-based view of the database schema and content. These two aspects of the infrastructure are measured separately.

The hidden levels of infrastructure—the server, its storage, and its network connectivity—are measured by processing speed, storage capacity, bandwidth, reliability, etc. The user interface aspects are measured by both quantitatively, through measures such as task completion time and number of user actions, and qualitatively, with approaches that capture concepts such as "usability" and "intuitiveness." These methods can be pulled from a broad range of techniques in fields such as human factors and human-computer interaction.

USER

The user is depicted at the far right of FIGURE 1 (E), along with his interface to the rest of the BDES system (perception, control), his assets (e.g., knowledge, experience, expertise, style), and his limitations (needs, gaps, constraints). While the user has been considered from a human factors and ergonomics perspective for several decades,[7] these classic fields consider only a passive or physical user interface. More recently, Pew edited a compilation on human-system integration (HSI)[8]; HSI focuses on inclusion of the user throughout the systems engineering design process, including user selection, training, performance, and safety. The first handbook of cognitive engineering—an interdisciplinary study focused on human/technology operation to achieve system-level goals—has just been published.[9] It is this emerging field of cognitive engineering that we must leverage to understand the user as a component in a BDES.

Treating the user as a component, we can describe her in a number of dimensions. Descriptive dimensions include any number of psychometrics (personality, intelligence, cognitive style, etc.), along with her skills[10], both intrinsic (what she's good at—perception, planning, decision making) and germane (what she knows from experience—tacit knowledge, strategy to solution, unstated constraints, inherent structure in the problem). We can measure user performance against dimensions including workload and situational awareness.[11] Descriptive dimensions are typically measured prior to an evaluation of the user working with a tool. Performance dimensions may be measured retrospectively, or through occasional interrupt protocols during work. The latter measurements are typically infrequent to avoid persistent interruption to the task at hand, and capture—at best—a coarse progression of the user as she works. Qualitative measures of performance, as provided by contextual inquiry methods[12], can be developed by third-party observers, and then refined with the help of the user. However, thinking of and measuring users in the same way as other components is, like the fields of HSI and cognitive engineering, very new territory with few best-practices for guidance. We will return to this issue, at length.

MISSION

The context of a BDES is the mission for which it is being applied (see FIGURE 2). The mission sets requirements on the BDES at a system level, and components must be integrated properly to meet those needs. The mission defines the task: the data to be exploited; the hypotheses to be tested; the desired precision of the final answer. The mission further impacts the user with time constraints, distractions, and incidental mission burdens. In order to measure the performance of a BDES, one must place it within the context of a mission with defined requirements, exogenous stresses, and targeted outcomes.

The mission is also the environment within which the integrated components of the BDES are activated. Neglecting hardware failures, during a mission, the data, analytics, visualizations, and infrastructure can be expected to perform according to specification. The user, on the other hand, brings a dynamic, innovative force to the BDES. Emergent behavior—beneficial or detrimental—can occur as a result of the user's insights. This is how we discover both answers and new questions within big data sets; this is the exciting opportunity that is big data exploitation.

COMPONENT-LEVEL MEASUREMENT IS NOT SUFFICIENT

There are myriad and well-established ways to measure each of the components that comprise a BDES; we have just scratched the surface in the previous summary. However, measurement at the component level is insufficient to describe system performance. The whole is greater than the sum of its parts and a set of individually optimized components—even a set that has worked well together previously—may be thoroughly inadequate for meeting the demands of a new mission or data set. This point cannot be overemphasized, as the user frequently has no ground truth for assessing, and may not have the skill-set for questioning, the plausibility of a result. The combinatorics of potential component integration failures span the space: mission-capable experts, but with new data; experts with a data set, but viewing it against different bases; a user with deep understanding of an algorithm, but inexperienced in its mission application. Critically, the user's experience with the data and the task at hand is entirely moderated by a set of tools that she is unlikely to fully understand. When such tools are used outside of their intended domain, emergent behavior (unexpected system states) will occur. This is true in complex physical, network, and socio-technical systems.[13] Without measurement at the system level there is no way of predicting whether the states that emerge will result in faulty conclusions, mission failure, or simply the inefficiencies of user confusion and frustration.

Consider, for example, the early failures of computer-assisted design (CAD) tools used by mechanical engineers as described by Petroski. He reflects upon the rapid revolution from slide rule to calculator to personal computer, and the adoption of CAD tools by knowledgeable engineers. As with BDES, CAD tools are "both blessing and

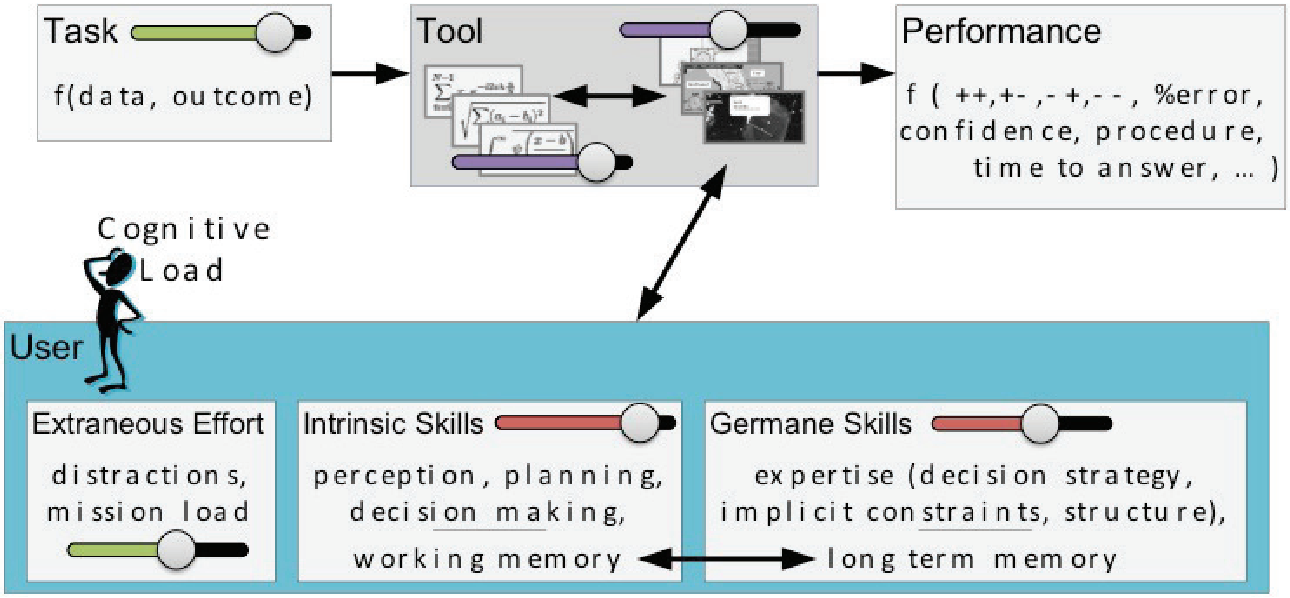


TABLE 1: *Summary of Descriptive and Performance Component-level Metrics for BDES Components*

curse for it makes possible calculations once beyond the reach of human endurance while at the same time also making them virtually beyond the hope of human verification."[14] Petroski notes that models can have errors, can be improperly employed, and can be incorrectly interpreted—true for both CAD and BDES. Such failures may be tolerable in BDES commerce or entertainment markets (targeted advertising and friending recommendations may be off-point)—but become critical in areas including defense, intelligence, and clinical decision support (where failures can cost lives).

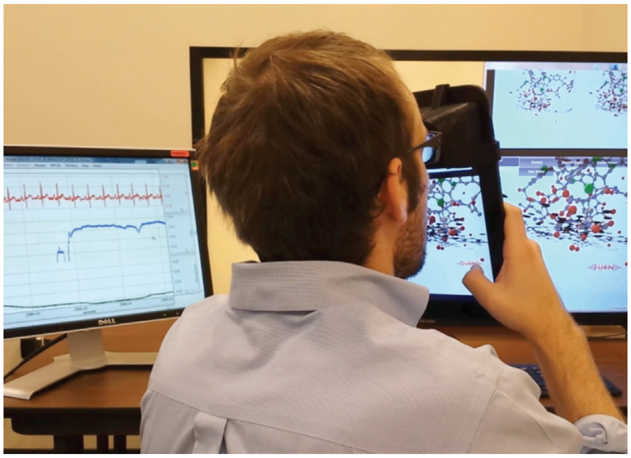
A TECHNICAL SOLUTION FOR QUANTITATIVE, NON-SUBJECTIVE METRICS

As described above, the present solution for addressing BDES performance has been at the component level. However, early CAD tools were well-conceived at the component level; failures occurred by neglecting the user and the mission, these system-level elements which are so important in operational use. Projecting forward, this places the burden on the user to understand what their own behaviors mean in context with the BDES and mission. Canonical techniques for measuring users are born of this perspective and seek to improve system functionality through one-time state measurement, questionnaires, or performance and response data once removed from task workflow. Such methods treat the user as a separate system. One cannot hope to capture a BDES's dynamic, emergent performance with such indirect or falsely bifurcated methods.

With traditional methods of infrequent or indirect measurement of the user, how can we hope, for example, to compare two user's experiences within a BDES, or across multiple BDES compositions (see FIGURE 3)? Every time we exchange a BDES component (including the user), we run the risk of failing to recognize that we have fundamentally changed systems, and making the same mistakes Petroski describes.

DEVELOPING A USER FRAME OF REFERENCE

Modeling the performance of BDES at a system level, with Nyquist-criteria sampling resolution, and incorporating the user as a component places new requirements on methods for measuring user behavior. User behaviors must be captured and quantized with sufficient resolution to enable modeling as sequential or synchronous inputs/outputs of the other system components. This is a radical departure from traditional approaches to measuring user behavior and performance, as described above—infrequent measurements of the user component-level state,



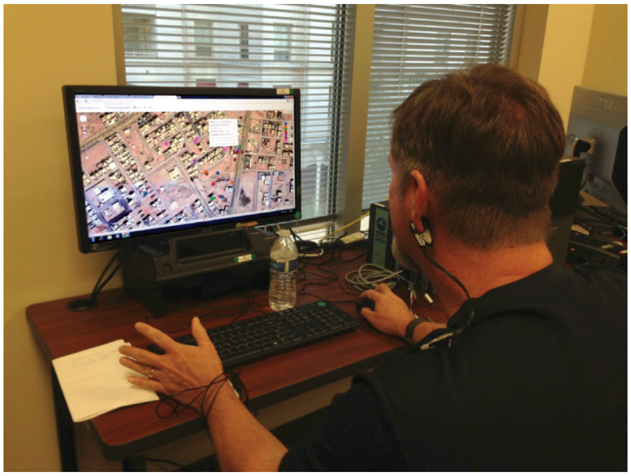


FIGURE 3: *Two snapshots of BDES under measurement. BDESs from the DAPRA XDATA program are shown, with different data sets, analytics, visualizations, infrastructures, users, and envisioned missions.*

3D stereo viewer developed by USC ICT: http://mxrlab.com *(top); note physiology waveforms on left monitor.*

Query and visualization components developed by Kitware, Inc. http://www.kitware.com *(bottom); note eyetracker unit (under monitor) and pulse plethysmography sensor (left ear).*

often aggregated and sometimes retrospective. So, how do we make use of users' behavior within the BDES such that it meaningfully relates to both the actions of other components and the holistic system? We believe this is question of context.

Context is a frame of reference for describing user behaviors within a system. We can think of each component as providing its own context—each BDES component has different purpose, each takes different inputs and different outputs, and transforms what is ultimately the same data into different views that feed into the net system output. The perpetual challenge in user experience is to imagine how the data and views are being interpreted by the user—this is the users' frame of reference. Without context we have no way of really knowing what their purpose is, how they interpret their inputs, what their outputs really mean and how they are transforming data into information. More importantly, we have no way of knowing whether those transformations—decisions, opinion, attitudes—are well informed or fundamentally flawed. We want to know the users' unconstrained frame of reference, and we want to be able to sample it at Nyquist-criteria resolution.

Users are incredibly complex components who bring inspiration to the use of BDES; we want them to explore and evaluate a new set of components without constraining their innovation, otherwise our metrics cannot project into operational use. That is, the frame of reference must not be pre-defined into a set of scripted steps. Instead, to meet these needs for sampling and interpretability and allow us to properly measure the user as a component within a BDES, we give context to users behaviors from the lens of other components, which are much simpler.

With this perspective, we can start to understand how the user and the other components work together dynamically. We must sample this data at the operational cadence set by the BDES components, the nature of the data, and the mission constraints (We invoke the classic concept of Nyquist-criteria sampling to denote this emergent resolution.). This data about the user, obtained passively, continuously, and without interruption of the task, is a new enabler of system-level performance assessment. In turn, it provides the basis for our approach to quantitatively measure the user to address systems-level behavior and performance questions. Questions may be framed regarding user behaviors with interface designs, regarding user activities within a class of BDES, and regarding user workflow states, enabling quantitative, non-subjective comparison across tools and missions.

APPLYING THE USER FRAME OF REFERENCE WITHIN CONTEXT

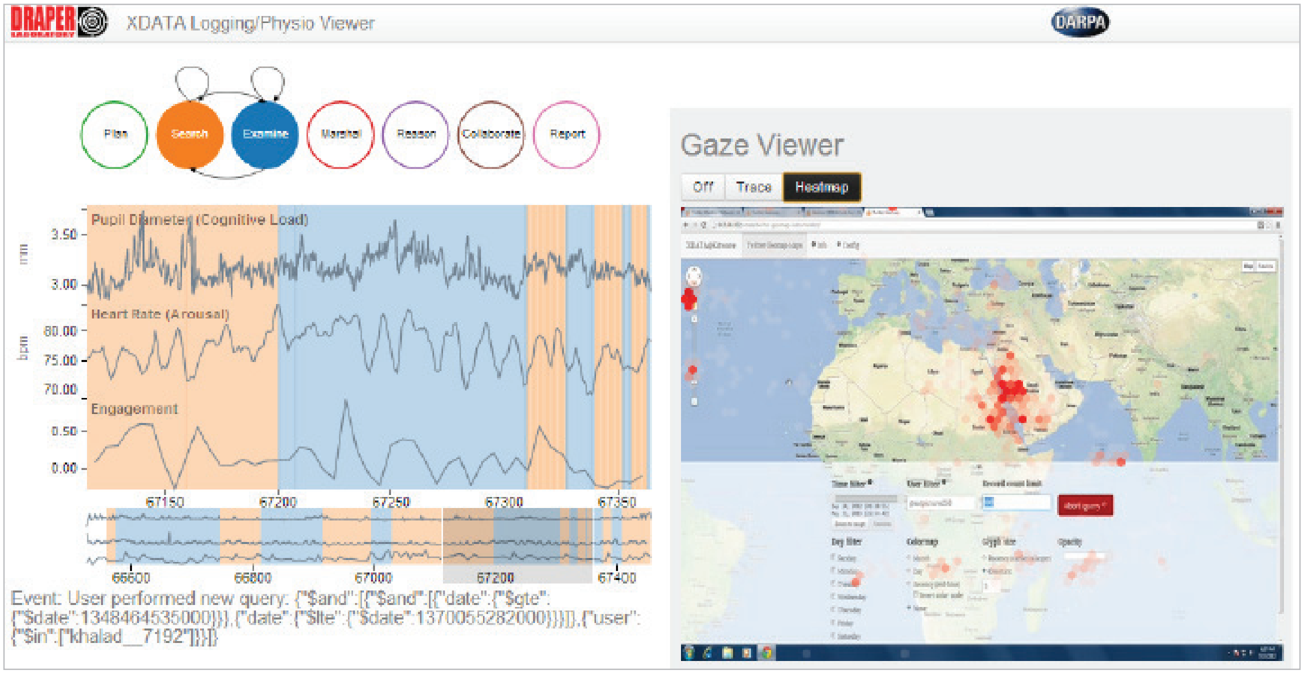
The interfaces to the other BDES components are the key to interpreting user behaviors and activities within a BDES, and an avenue for effectively measuring the user in mission context, quantitatively and non-subjectively. Capturing the context requires thoughtful consideration on how user behaviors (e.g., mouse clicks, keystrokes, gestures) map to a hierarchy of activities (e.g., query, filter, pan/zoom/rotate) within the BDES and mission workflow states (e.g., plan, search, view, organize). This does not assume a hypothesis on the users' intentions, nor that each behavior (e.g., click) is meaningful in its own right. Instead, the intent is to provide an activity- and state-based frame of reference for user behavior, leveraging a mature understanding of the BDES and mission space. The result is a context map, a simple lexicon, which interprets what a user does in context of the BDES and mission across time. This approach provides a means for sampling user behaviors directly, in the native language of the BDES, with a sampling that meets the cadence of the rest of the components.

Context mapping provides a basis for providing relevant data about the user; each behavioral data point adds little inferential value alone. For example, this mapping between user behaviors, BDES activities, and workflow states is agnostic to user intent, which is important for testing a number of critical hypotheses: is the user receiving the outputs she wants given the inputs she provides; do periods of inactivity mean that the user is thinking about the information he is looking at or that he's become distracted; are periods in which the user accesses a wide range of system functions indicative of excited exploration of a new lead, or incredible frustration? However, the aggregated vector of contextually-situated time series data can be subjected to the same kinds of analytic approaches that make event-related experimental designs incredibly useful for inference. Using the data within the time series vector provides reliability for the analysis, analogous to a low-pass filter for removing noise. For example, systematically categorized, diverse time-series data enable Markovian state modeling at the level of cognitive style, which can be used to make inferences about when users' strategies in analyzing data shift, and what those shifts look like in the native space of system activities. Theory-driven approaches can identify, with frequentist or probabilistic statistics, whether some distribution of system activities indicates transitions in workflow states of interest (i.e. hypothesis testing, data exploration). We can find systematic patterns that reveal the users' frame of reference—the data provided by the context mapping exercise gives a means of interpreting resulting metrics in the context of system performance. In turn, this provides a means for addressing whether all of the system components are working harmoniously.

APPLICATION ON THE DARPA XDATA PROGRAM

As part of the DARPA Information Innovation Office (I2O) XDATA program, Draper is currently developing broadly applicable metrics for assessing system-level value in the context of BDES. Our approach defines BDES Value as the ratio of Task Performance over Cognitive Load. Loosely, "how good" is the answer, moderated by "how hard" was it to obtain. Unlike component level benchmarking or other performance measurement, BDES Value enables selection among the sufficient system-level solutions that meet the necessary condition. The necessary condition is imposed by the mission, and may include the required accuracy of the answer, within a limited time. The best sufficient solution will meet those constraints without unnecessary burden on the user. Given two satisficing solutions, it is most robust to select the system that requires the least cognitive load, and/or the least time to completion.

We have developed an Analytic Activity Logging API for instrumenting software components. The API is designed to support the behavioral and physiological assessment (including eye tracking) of users as they interact with XDATA tools, in service of building a better understanding of how analysts make use of these tools. Through the use of a common activity logging API, we can ensure that the combined XDATA team generates uniform logging of user actions across varied analytic and visualization products. Since activities can vary greatly across the various components, there is a challenge to capture the user's activity with enough specificity to characterize the specific action within an application (that is, a set of integrated components), while also describing the behavior in a general way so that user activity logs can be compared across applications. To accomplish this, user actions are described at 3 levels of abstraction: action description, activity, and workflow state. Workflow states provide a means of grouping actions across applications, activities provide a means to group actions within applications, and the action description specifies the detail of each action. The workflow states are defined based on a nominal analyst workflow and further informed by contextual inquiry with over 30 professional



Event: User performed new query: {"$and":[{"$and":[{"date":{"$gte":{"$date":1348464535000}}},{"date":{"$lte":{"$date":1370055282000}}}]},{"user":{"$in":["khalad__7192"]}}]}

FIGURE 4: *Analytic Activity Dashboard, showing action, activity, and state. User physiology overlaid on event logs shown on left. Kitware, Inc. component with gaze vector information shown on right,* http://xdata.kitware.com/examples/flickr/.

analysts, as well technical expertise in the underpinnings and operation of BDES.[15]

We have also developed a logging server to receive and store the activity logs from multiple components simultaneously. The server is written in Java Script, and leverages Node.js and MongoDB. The API, helper libraries for Java Script, Java, Python, and C# development languages, and server code are available under the Apache License, Version 2.0, from *https://github.com/draperlab/xdatalogger*.

In order to aid in the analysis of the logging activity in relation to the recorded physiology, we have further developed an interactive dashboard for exploration of behavior, activity, and state data, compared to user response (FIGURE 4). The logged data are overlaid onto the time series physiologic signals (waveforms, on the left, and gaze vector data, on the right) to view how one changes in relation to the other. This screen shot depicts a test of a set of Search and Examine components from the Kitware team. All dashboard components are developed using d3.js, an interactive JavaScript library that binds data to HTML objects.

CONCLUSION

Lessons learned from the discipline of systems engineering make clear than when designing and testing a system, it's imperative to ensure that what is being designed and tested is a system of interdependent components—not just the components themselves. In hindsight, this tends to be obvious, but even the most thoughtful developer may be unwitting of the roadmap of an evolving mission such as is enabled by big data. Further, it is still not common to recognize the user as one of the most interdependent of these components.

BDES require user-centered design processes and thoughtful integration. Complexity is inherent in big data—and its volume, velocity, variety, and veracity—and the other components of the BDES are the only means the user has to make sense of that complexity. In such tightly coupled systems, proper measurement at the system-level, framed against the mission, is key to understanding performance.

Without system-level metrics, emergence of unanticipated states—dynamic changes in how data maps to understanding, for example—will be the rule, not the exception. If the user and their mission are not well integrated with the rest of the BDES, the user may be led to erroneous conclusion or down paths of inquiry without end. These are the risks of working with big data, or in any virtualized problem space where the user has lost her sense of intuition. However, if all the pieces of the BDES fit together elegantly, the user just might get that glimpse of the big picture that is only possible with big data, and makes the emerging use of BDES so exciting.

ACKNOWLEDGEMENTS

The project or effort depicted is sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL).

BIOGRAPHY

Dr. Jana Schwartz is a Principal Member of the Technical Staff at Draper Laboratory and Group Leader for Human-Centered Engineering. She has interests in human/system collaborative operations, complexity theory, and the use of real-time psychophysiological response for closed-loop performance, and is the PI of Draper's XDATA team.

Dr. Joshua Poore is a Senior Member of the Draper Laboratory Technical Staff and experimental psychologist. He has applied this work to developing comprehensive metrics for user states (i.e. immersion, proficiency) within human computer interfaces, including virtual environments, simulations, gaming, and analytic tools.

Ms. Emily Vincent is a Senior Member of the Technical Staff in Human-Centered Engineering at Draper Laboratory. She has led human-system collaboration and software tasks for intelligence community and SOCOM programs, including the development of the Next Generation BAO Kit roadmap and TacDroid™ prototype.

Mr. David Reed is a Member of the Technical Staff at Draper Laboratory. He has interests in data visualization, human/system collaborative systems, signal processing, machine learning, and data science.

NOTES

[1] Jeffery Dean and Sanjay Ghemawat. "MapReduce: simplified data processing on large clusters." Communications of the ACM 51, no. 1 (2008): 107-113.

[2] Bo Pang and Lillian Lee. "Opinion Mining and Sentiment Analysis." Foundations and Trends in Information Retrieval 2, no. 1-2 (2008): 1-135.

[3] Willie Walker, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, and Joe Woelfel. "Sphinx-4: A Flexible Open Source Framework for Speech Recognition." (2004).

[4] Paul Zikopoulos, Krishnan Parasuraman, Thomas Deutsch, James Giles, and David Corrigan. Harness the Power of Big Data The IBM Big Data Platform. McGraw Hill Professional, 2012.

[5] Saptarshi Guha, Ryan Hafen, Jeremiah Rounds, Jin Xia, Jianfu Li, Bowei Xi, and William S. Cleveland. "Large Complex Data: Divide and Recombine (D&R) with RHIPE." Stat 1, no. 1 (2012): 53-67; Sungpack Hong, Sang Kyun Kim, Tayo Oguntebi, and Kunle Olukotun. "Accelerating CUDA graph algorithms at maximum warp." In Proceedings of the 16th ACM symposium on Principles and Practice of Parallel Programming, pp. 267-276. ACM, 2011; Pinar Donmez, Jaime G. Carbonell, and Jeff Schneider. "Efficiently Learning the Accuracy of Labeling Sources for Selective Sampling." In Proceedings of the 15th ACM SIGKDD international conference on Knowledge Discovery and Data Mining, pp. 259-268. ACM, 2009; Vikash Mansinghka, Charles Kemp, Thomas Griffiths, and Joshua Tenenbaum. "Structured Priors for Structure Learning." arXiv preprint arXiv:1206.6852 (2012).

[6] Leland Wilkinson. The Grammar of Graphics. Springer, 2005.

[7] Etienne Grandjean. Fitting the Task to the Man: a Textbook of Occupational Ergonomics. Taylor & Francis/Hemisphere, 1989.

[8] Richard W. Pew and Anne S. Mavor, eds. Human-system Integration in the System Development Process: A New Look. National Academies Press, 2007.

[9] John D. Lee, and Alex Kirlik, eds. The Oxford Handbook of Cognitive Engineering. Oxford University Press, 2013.

[10] Jan L. Plass, Roxana Moreno, and Roland Brünken, eds. Cognitive Load Theory. Cambridge University Press, 2010.

[11] Sandra G. Hart and Lowell E. Staveland. "Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research." Human mental workload 1, no. 3 (1988): 139-183; Susan Rubio, Eva Díaz, Jesús Martín, and José M. Puente. "Evaluation of Subjective Mental Workload: A Comparison of SWAT, NASA-TLX, and Workload Profile Methods." Applied Psychology 53, no. 1 (2004): 61-86; Mica R. Endsley. "Situation awareness global assessment technique (SAGAT)." In Aerospace and Electronics Conference, 1988. NAECON 1988., Proceedings of the IEEE 1988 National, pp. 789-795. IEEE, 1988.

[12] Hugh Beyer and Karen Holtzblatt. Contextual design: defining customer-centered systems. Access Online via Elsevier, 1997.

[13] Donna H Rhodes, Adam M. Ross, and Deborah J. Nightingale. "Architecting the System of Systems Enterprise: Enabling Constructs and Methods from the Field of Engineering Systems." In Systems Conference, 2009 3rd Annual IEEE, pp. 190-195. IEEE, 2009; Barabási, Albert-László, and Réka Albert. "Emergence of Scaling in Random Networks." Science 286, no. 5439 (1999): 509-512; Nathan Eagle and Alex Sandy Pentland. "Eigenbehaviors: Identifying Structure in Routine." Behavioral Ecology and Sociobiology 63, no. 7 (2009): 1057-1066.

[14] Henry Petroski. To Engineer is Human: The Role of Failure in Successful Design. Chapter 15. New York: Vintage books, 1992.

[15] Joseph MacInnes, Stephanie Santosa, and William Wright. "Visual Classification: Expert Knowledge Guides Machine Learning." Computer Graphics and Applications, IEEE 30, no. 1 (2010): 8-14.