Human Machine Interactions: Velocity Considerations

Joseph Cottam(✉), Leslie M. Blaha, Kris Cook, and Mark Whiting

Pacific Northwest National Laboratory, Richland, WA 99352, USA
{joseph.cottam,leslie.blaha,kris.cook,mark.whiting}@pnnl.gov

Abstract. Measuring change is increasingly a computational task, but understanding change and its implications are fundamentally human challenges. Successful human/machine teams for streaming data analysis effectively balance data velocity with people's capacity to ingest, reason about, and act upon the data. Computational support is critical to aiding humans with finding what is needed when it is needed. This is particularly evident in supporting complex sensemaking, situation awareness, and decision making in streaming contexts. Herein, we conceptualize human/machine teams as interacting streams of data, generated from the interactions that are core to the human/machine team activity. These streams capture the relative velocities of the human and machine activities, which allows the machine to balance the capabilities of the two halves of the system. We review the known challenges in handling interacting streams that have been distilled in computational systems. And we use this perspective to understand some of the open challenges to designing effective human/machine systems that support the disparate velocities of humans and machines.

Keywords: Big data · Human-machine interaction
Interactive streaming analytics · Visual analytics

1 Introduction

Despite the availability of "big data," people remain effective processors of only small data. Human perceptual and cognitive capacities remain constant, including the effective limits on our working memory [1–3], limits on our attentional capacity [4,5], and often limited bandwidth information processing capacity [6–8]. People are also susceptible to a number of cognitive moderators that reduce our ability to process information quickly and without error, such as fatigue, stress, or cognitive overload [9]. We must rely on interactive interfaces to supply the functionality to balance the difference between small-data processing people and big-data processing machines.

The implications of this reliance on interactive interfaces to moderate the speeds of human and machine activity are two-fold. First, all the primary responsibility for processing and handling data falls to the machine. Machines can be

D. D. Schmorrow and C. M. Fidopiastis (Eds.): AC 2018, LNAI 10916, pp. 43–57, 2018.
https://doi.org/10.1007/978-3-319-91467-1_4

built to operate at the speed and capacity needed to handle the velocities of the data, the algorithms, and the user inputs. Machine intelligence, including complex algorithms and artificial intelligence, can be employed to determine how to execute processes and to provide solutions under dynamic system demands. Second, the design and engineering of those systems falls to humans. Human intelligence must determine the requirements of both the computing and human information processing elements, complete with user requirements, variable working environments, and changing goals.

The purpose of big data and computational resources is to be helpful to ourselves: we gather data and employ resources to solve human problems. With advancing machine technologies, the interaction between humans and machines has changed shape. Smaller, more efficient, and more powerful machines have both quantitatively and qualitatively changed what we try to compute on and the expected output of that computation. The development of *deep learning* is a recent example of this: it became computationally feasible to execute large-scale analysis (including higher speed networking, greater storage density, and improved compute power) that was previously unreachable. Quantitatively, more operations per second and larger data were economically available. Qualitatively, pervasive data collection resources provided richer data than had been available in the past. Collectively, these abilities are enabling advances in life-critical applications, such as self-driving cars and advanced health-care. Importantly, a major contributor to the success of many deep-learning efforts are the data gathered through new *interaction* patterns found in crowd-sourcing information (e.g., Mechanical Turk, Re-captcha) [10].

There are many conflicting definitions of "big data" but the majority share variants on three characteristics V's: volume, variety, and velocity [11,12]. (Other definitions add characteristics such as veracity, volatility, variability, validity, and value. It is beyond our scope to explore the necessity, correctness, or implications of each.) Each characteristic carries different constraints on human/machine teaming and system engineering. Velocity is of particular interest at present because it is a source of fragility in human/machine teams. Velocity of big data refers to the speed at which data is generated, recorded, or refreshed. Because machines are often involved in the production, data velocity is implicitly related to the speed at which machines can process some source data. Importantly, the producing machine and the analyzing machine are often distinct.

A human/machine team must also consider the velocity of the human: people's capacity to ingest, reason about, and act upon the data. Considering velocity from both perspectives, time is a shared but relatively inflexible medium. There is little we can do to change our experience with time: it does not wait or stretch or save for humans or for machines. However, humans and machines experience time and events over time on very different scales. Indeed, they are even measured in different units, with meaningful human activity measured in wall clock time in tens of milliseconds to years [13] and meaningful machine activity happening in system time down to the nanosecond. Consequently, apparent velocity versus relative processing speed has significantly different implications for human/machine teams. Regardless of how fast things are happening or the measurement scales, "velocity" implies change over time, which means sequencing and distributions of events become meaningful in the context of time. Specifically, time and sequence restrict the analysis and force constraints on human reasoning.

Velocity, and particularly the difference in velocity between machines and humans, is an important consideration for the development of systems performing interactive streaming analytics at scale. Interactive streaming analytic systems, schematically diagrammed in Fig. 1, are human/machine systems that strive to enable human decision making on streams of data in (near) real-time. The goal is to produce correct, useful, and timely interpretations of the world, where a human's experience of the world is mitigated by the machine representations of the data streams. Note that when we refer to streams, we are referring to continuing generation and collection of data where the values are changing while being observed. Stream analytics steer data collection, summarize information, and manage events occurring in the streaming data. Such analytics may capture dynamic query updates and employ interactive interfaces to provide human-digestible information to support intelligent decision making. Such systems provide the joint human/machine intelligence needed to perform real-time systems monitoring (e.g., power grid control stations or flight control operations), continuous security screening, or mission command and control. Putting users into the streaming analytics process requires that interactive interfaces effectively match both the velocity of both human cognitive inference and interface interactions as well as the machine-derived velocity of updates to analysis and visual representations. Thus, interactive streaming analytics at scale requires an interface to mediate the difference between humans and machines to facilitate meaningful interactions.

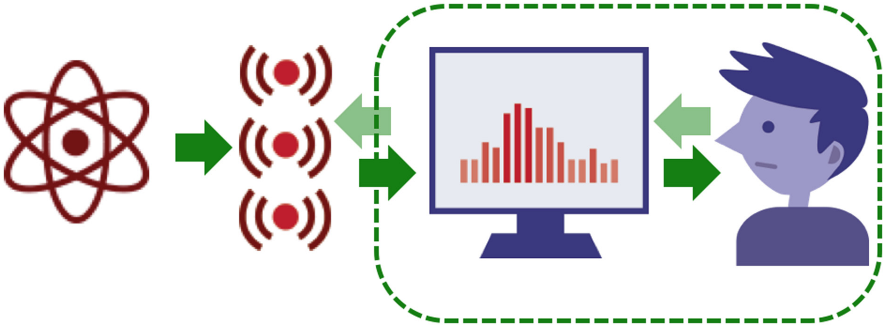

Fig. 1. Human/machine teaming for interactive streaming analytics at scale. The human/machine team is contained in the green box, with the interactive user interface mediating the differences in the velocity of the user activity (right side) and the velocity of data streaming in from the world (left side). (Color figure online)

Despite the need for interactive streaming systems in response to increasing data availability, we have rarely succeeded in developing effective solutions to this problem, let alone general solutions. One reason for a lack of general solutions is that we lack a complete understanding of the conceptual and design spaces and how they interact. This trade-off space captures the challenges of developing systems for large scale, streaming data and the related challenges and opportunities emerging from the cognitive capabilities of human operators working on streaming data. In this paper, we argue that designing and engineering for human/machine teaming in interactive streaming analytics or other dynamic operational environments is critical for finding effective solutions to balance the velocity of humans, the velocity of computers, and the velocity of the phenomena to which the human/machine must be responsive. We believe the key is emphasizing the interactions, because interaction data processing can be handed to the machine, alongside other data streams, and the humans can focus on designing effective interfaces to facilitate those interactions. From this perspective, human/machine interactions create interacting data streams, and leveraging known approaches for handling data streams can help us shape the questions about how to design effective human/machine teams to operate in dynamic, streaming conditions. Thus, we explore how reflecting on velocity shapes what designers need to consider about both the human and the machine to make an effective team.

2 Human/Machine Teams

The core of any team is interaction. Interactions provide the opportunity for team members to influence each other, and become a joint entity that is more than the sum of its parts. The information passed across a user interface (UI) is therefore crucial to understanding how a human/machine team acts. Fluid human/machine teaming is particularly critical for interactive streaming analytics systems that seek to put a human into the loop for real-time analysis [14].

Algorithmic, hardware, and cultural changes have enabled computer assistance in problem solving. Humans and machines form a team with distinct but complementary capabilities for analytical tasks. Humans are masters of context: capable of bridging gaps between disparate datasets and making heuristic judgments within datasets. In most cases, some vital information people work with cannot be easily encoded for computational reasoning. Humans are also able to act on information in ways distinct from computers, interfacing with the world outside of the analytical environment. The human side of the team brings the motivation for problem solving and is the ultimate arbiter of the goals to pursue.

Computer capabilities are often engineered to be complementary to human capabilities. Machines provide raw capacity, repeatability, and a certain flavor of efficiency. Computers handle large amounts of data at high speeds, they provide repeatable processes, and tools for verifiability and precision. In a streaming context, these machine capabilities are invaluable to human problem solving.

However, human and machine capabilities are not automatically cohesive. It takes attention to merge the two into a team. This is doubly true in streaming context, because interactions need to focus on timely data analysis instead of learning system behaviors and functionality. It is desirable in this context to enable the system to simply observe the patterns of user interactions, and to use that data in meaningful ways to help shape the human/machine team into the cohesive team that can effectively perform the tasks of interest (e.g., [15,16]). The latter statement, however, encompasses a broad set of open challenges in how to use data in meaningful ways.

A number of advances in developing cohesive teams may be resolved by establishing better models for each teammate. If computers were better equipped with better human models, they would be able to more able to operate as a team member. Current computer-hosted models of humans are typically static and highly reactive (consider, most user interfaces essentially model likely "next" interactions or expected constraints on interactions). Improving these models could enable more proactive machine roles. Humans with better models of the machine can more effectively communicate with it and interpret the results. The interaction patterns between the humans and machines offer a key measurement of the interacting agents that can be used to establish and inform such models.

2.1 Human Input

Human inputs to a machine interface are naturally modeled as a stream. They are time varying, potentially unbounded, and not inherently repeatable [17,18]. Streaming data analysis systems that incorporate user input in a non-trivial way therefore must deal with the intersection of the user input with other data streams. These issues extend beyond general multi-stream issues discussed in Sect. 4 (though many of those issues should also be considered when dealing with human input).

Our interest is in human/machine teaming systems that have non-trivial user streams. For our purposes, "non-trivial" user inputs are those that influence how future items in other streams may be processed. In some systems, user inputs only inspect current items available in the user interface (UI). That inspection does not influence future system behaviors, unless mixed-initiative system approaches are engaged to indirectly influence future behavior through machine steering. Imposing a filter is perhaps the simplest non-trivial input: future items may or may not be displayed. This interaction is simple because it is deterministic and essentially algebraic in nature. More complex interactions include training examples through semi-supervised learning [19,20], semantic interactions [21], and interaction sequences that might indicate foraging or other exploratory analytic processes in provenance modeling. Conceptually the difference between trivial and non-trivial inputs can be roughly approximated thus: in a client-server system, a non-trivial user input is one that crosses from the client back to the server, while trivial inputs stay within the client [22].[1]

[1] This is a heuristic modeled on server-client designs. A sufficiently thin client will not be able to do simple inspection on its own, and a sufficiently capable client may do machine learning.

Practically, non-trivial user inputs may come in the form of keyboard actions, mouse clicks, touch inputs, or verbal communication, based on the design of the system. Although input details influence the exact velocity of the system, our stream-based model is agnostic to them because each will have a sequence and human-scale velocity.

2.2 Machine Input

Machines contribute multiple different kinds of input into a system. In addition to being integral in the data collection pipeline, machine actions around analytics and UI can also define a stream. Consider, for example, mixed-initiative computing systems. Tracking of user inputs is a hallmark of mixed-initiative systems for visual analytics [23]. In mixed-initiative systems, user interactions serve to steer and inform the machine analytics. Machine actions then make recommendations to the user for future interactions (an example of this recommendation workflow style is the Active Data Environment [24]). This approach creates a series of machine actions that influence the information and options presented back to the user. This is the machine interaction stream.

Semantic interactions were introduced to describe UI interactions that suggest particular interpretations to the machine because they occur in the context of a known interface metaphor [21,25]. Broadly, machines might leverage semantic interactions to understand the user's mental model and sensemaking process [26]. In many current mixed-initiative systems, the semantic interactions are interpreted as they occur through the course of analytic session; they are not necessarily revisited or reinterpreted, even if the analytic provenance is recorded. The machine reactions to semantic interactions are not usually directly analyzed, but they also reflect a component of the machine interaction stream. Conceptualizing machine interactions as a stream will capture all the key characteristics of the velocity of machine intelligence, which can be aligned with the velocity of human interactions and with the data ingestion and processing velocities.

3 (Partial) Streaming Solutions

Stream processing algorithms and systems have been an active area of research for decades. This existing work establishes a foundation for machine processing of streaming data at a variety of velocities. This section summarizes machine-processing oriented observations about streams.

3.1 Time & Sequence

A stream of values arrives over time, implying a sequence of values that arrive over time. Therefore, stream algorithms must work with changing values or a growing collection of values over time. This is distinct from traditional time-series analysis. In time-series analysis, time itself is a part of the data being analyzed. Often a full time series will be available at once, so the sequence of values is not integral to the analysis strategy. In streaming analysis, time *may* be part of what is being analyzed; but sequence of values *must* be part of the analysis strategy (either to ensure it does not impact results or to employ it to improve results).

3.2 Sequence

A stream implies that values are presented over time. However, the presentation order does not need to correspond to their production order. Out-of-order appearance is common in many real world cases. A practical example is in the Transmission Control Protocol (TCP), which includes specific provision for out-of-order delivery of (streamed) packets. In a trivial case, production may be the iteration of values from database in reverse temporal order. As these two examples illustrate, a sequence may be irregular or regular and sequence deviations may be incidental or deliberate. Regardless of its form, origin and severity of sequence issues, if present, must be addressed.

3.3 Data Rates

Many streams have “too much” content to fully capture or practically store the entirety. In other words, the data rate may be too high to handle, either computationally or cognitively. Issues of frequency directly influence the scaling of a system computationally, and ability of humans to act on computational results. Choices must be made to determine sampling rates that are practical according to sensor limitations, that capture the meaningful events on the stream, and that enable the “right” amount of data to be analyzed and/or stored.

3.4 Cadence

Stream cadence may be highly regular or prone to bursts. Time-windowed stream volume may be variable as well. These have technical and interface implications.

On the technical side, cadence directly impacts resource allocation strategies. Regular cadence enables efficient resource management. Irregular cadences can be mitigated with buffering strategies (e.g., burst buffers on HPC machines, Cloud Flair for website requests) that smooth out the stream, trading variable volume for variable lag. Another option for handling variability is to simply “drop” stream content when it exceeds the design threshold (as is done in the ethernet networking protocol).

In the interface, stream cadence influences how updates are communicated. Whether irregular or regular, the cadence of updates itself may be part of the signal of interest. Displaying the distribution of updates can alleviate the cognitive burden of wondering about update frequency. The interface design should also consider the expected cadence range when implementing updates to minimize noise in the display.

3.5 Time Spans vs. Time Moments

The content of a stream may have drastically different *semantics*, even if the content is substantially similar. Returning to the sensor example, there are many valid configurations. A sensor may take measurements at given intervals and report each. It may take measurements at given intervals and only report those that differ from the prior. In the first case, the measurements are for moments in time, in the second reported measurements are for spans.

4 Crossing Streams

Many issues are compounded when multiple streams are brought together [22]. Stream velocities may differ in their rates, dimensionality, and variability. The contrast between different streams introduces many new characteristics, like cross-correlations or asynchrony, that must be dealt with in both analysis and interpretation.

4.1 Relative Skew

Audio and video are common streams encountered in a time-correlated fashion. YouTube, television, and films all rely on small temporal skews to maintain the illusion that the elements of the images are in fact the sources of the sounds. If the skew between the two streams becomes too large, the illusion of causality is broken. Whenever combining streams, issues of skew need to be considered. Do the streams travel similar paths, or is there an inherent lag between them? If they are produced together, do they travel together?

4.2 Multiple Resolution

Data from different sources often do not match in temporal resolution. This problem is often masked by a time-stamp convention that records all time as offset seconds (such as Unix timestamps). Regardless of the granularity of the measurement, the granularity of the time-stamp is driven by the accuracy of the available *clock*. Detecting that different resolutions are at play is the first problem.

The second problem is working with values recorded from different resolutions. For values with well-defined summarization properties, this can be straightforward operationally, but still requires users apply knowledge of context. For example, if one stream reports daily temperature and another reports hourly temperature from another location, what is the best way to mix the two? Maximum, minimum, averaging, random selection, and periodic selection can all be trivially applied to the more frequently reporting stream, but a key question is: which will provide the best comparison to the less frequent stream? Knowledge of how the slower stream is created likely dictates what is "best". Consequently, combinations are *operationally* simple while remaining *contextually* difficult.

4.3 Stream Interdependence

Streams may not be independent. Consider a UI for an industrial control application. The display shows the results of a stream of a data, capturing the current state of the system. The user inputs are a stream back to the system that influences future values. In other words, human inputs are in response to events on task streams. They can also influence what happens later on that task stream.

5 Implications for Design of Effective Human/Machine Teams

We have conceptualized human/machine teams as systems designed for non-trivial interactions at the core, and that those non-trivial interactions produce streams of data. Thus, we have a way to capture the velocity of human and machine behaviors, as well as the relative impact of those velocities on each other. All human and machine data streams can be analyzed by the machine, meeting its responsibility for handling all the data. We next consider some of the system behavior considerations that are known to influence the effectiveness of human/machine teams. In light of adopting the perspective of human/machine teams as interacting data streams, we elucidate some implications for these factors, particularly with respect to the management of differing velocities. And we identify open research questions for the human-centric system design process.

5.1 Different Working Speeds

The working speed difference between humans and machines is an obvious point of both conflict and complementary abilities. If every shift of machine attention were presented to a human to interact with, the machine would be left largely idle. Human relative slowness provides a mediating reservoir of attention. Thus, system designs can use the slower speed of humans to aggregate and synthesize machine activities and outputs before presenting to the user. Using the interacting data streams to be aware of the human attention availability and engagement, the machine can manage this timing adaptively.

5.2 Communication and Reconciliation

How do team members communicate goals, findings, and the need for status updates with each other? There are issues of interruption, non-computationally represented information, and changing priorities to be considered. In verbal communication between people, this often seems straightforward. Yet, between humans and machines, the communication barriers often seem insurmountable, despite advances in spoken-communication systems (like Siri and Alexa). Multi-modal and bi-directional communication is enabled through interacting streams. As the world and streaming data evolve over time, both the human or machine may communicate a new context. The interacting stream capture the evolving mental models or state of the agents. But a key open question remains: how does the team's shared state evolve?

5.3 Workflows and Task Allocation

Division of labor between human and machine intelligent agents is informed by a number of system attributes, including the nature of the tasks and the specific capabilities of the computational resources. Interacting streams capture fluctuations in working cadence, as well as information about what the machine and user are working on through semantic and mixed-initiative interactions. This enables adaptive and dynamic task allocations. It may also enable effective multitasking in humans, who often think they are good multi-taskers but rarely are [27,28]. Capturing the relative machine and human cadences further raises opportunities to adaptively and effectively interrupt current activities. That is, the streams may inform the time points at which can each teammate might usefully provide input to the other, and in what form in which that interruption might be most influential.

5.4 Trust, Verification, and Uncertainty

A number of open questions exist around how to ensure that users trust and rely on the machine intelligence and autonomous system elements. How do you present information at the right level of detail to be appropriately considered? How do you develop an appropriate appreciation in the human of the machine's abilities, without introducing more work than the machine or human is able to assume? What are the levers and limits of interrogation that a human can employ against a machine? What does it mean for a machine to be certain? How much of a model of the human can the machine effectively employ in building the relationship? How complete a model does the human need of the machine? Designs based on interacting streams may help to address a number of these questions, and extensive research is needed to understand the relevant of design options that facilitate human and machine trust and reliance.

5.5 Perceived Autonomy

We apply computational machinery to problems humans want to solve. However, machine capabilities are expanding and with them comes an expectation of greater autonomy, meaning adaptive system behaviors to manage information and operate on data without direct human supervision. For many existing interactive systems, machine tasking for non-routine tasks is largely under the control of the human operator. If interaction streams can inform models of user intent and need, how well can we identify ways in which the machine make take the lead? How much work is *initiated* by the machine? How many resources can it apply to a task before asking for human input?

Autonomy deals with the adaptive ability to initiate lines of inquiry (autonomy in the large) and the adaptive ability to execute tasks without additional input (autonomy in the small). Both forms carry benefits and pitfalls, and have implications for the velocity of the related machine behaviors. Autonomy in the large provides the appearance of an able assistant (like Jarvis from Iron Man or the Doctor from Star Trek's Voyager) or a potential replacement. In a more contemporary context, contextually aware advisors such as spelling/grammar check, auto-complete in web-search or the Active Data Environment [24] demonstrate a level of machine autonomy. In potentially autonomous machines, how much work "on their own" is the owner comfortable with? How much do they want to direct themselves? Is it just "spare cycles" that the machine has for its own inquiries, or can it use a baseline percentage?

In the small, machines that require constant attention can be an annoyance and impediment rather than a benefit. Indeed, poor interaction design can lead to a lack of autonomy for all team members as each blocks the other's progress. However, when context dictates the proper resolution of ambiguities, it may be better to receive inputs than to simply guess. Humor around auto-correct is derived from failures when there is too much autonomy in the small, while dialog box fatigue that leads to everything being blindly accepted and installed is a failure of too little autonomy.

5.6 Influence

Machine autonomy implies the active metaphor for instruction is not command, but influence. This influence can be inferred from the interaction of the user by having the machine interpret the interactions in the context of computational models of user intent. This is an indirect means of influence. At the opposite end of the spectrum is more direct influence, such as allocating computational resources to specific tasks or placing explicit priority markers. These are methods for a human to influence a machine's behavior. Machine influence of human behavior may be observed in the order that options are presented in and in the intrusiveness of interface elements. The salient aspect of influence is that it is indirect and that it preserves choice in when and how actions are taken.

6 Case Study: Test Harness

Considering interaction as a core consideration of human/machine for streaming data analysis leads to different considerations in system design than otherwise. Our experience designing and implementing test infrastructure for interactive streaming analytics is illustrative. Although human-in-the-loop data analytics are desirable in many settings, humans contribute variable behaviors to the team. This variability introduces novel challenges to system testing and evaluation.

In traditional software testing, algorithmic correctness and overall robustness are the principle consideration. For algorithmic correctness, systems are tested to see if inputs produce the expected outputs (or for stochastic systems, an output in an acceptable range acceptably often). A system state may be supplied artificially through mocks or stubs [29] to simulate the context developed over longer executions. Acceptable system robustness is assessed through some form of stress testing, checking for resource usage and the ability to remain operational (or degrade gracefully). Inputs in either case are typically supplied by the system

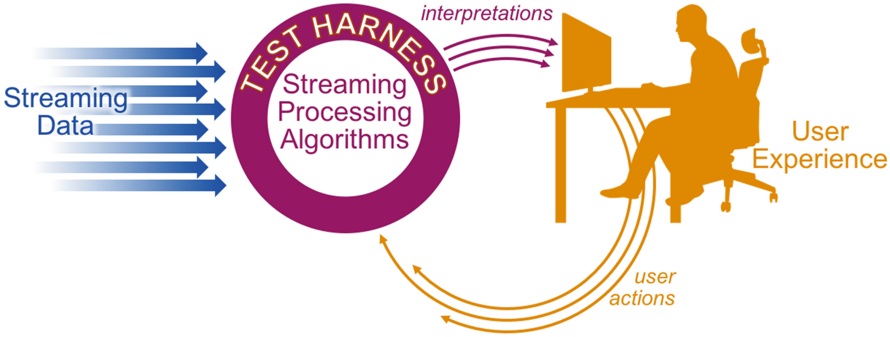

Fig. 2. System-level context for test harness.

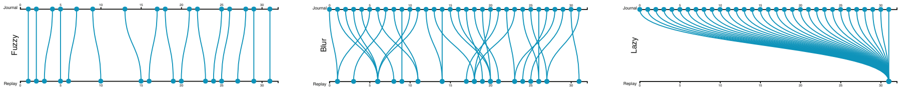

Fig. 3. Three test patterns. Fuzzy (left) tests for minor changes in timing. Blur (center) tests for changes in sequence. Lazy (right) tests if the user input needed to be provided mid-stream at all.

designer, possibly with guidance from a potential future user. Other properties that are commonly tested for include resource utilization, system responsiveness, data security or feature regression. Each of these properties is valuable in a useful system, but the mock/stub/etc. architecture often glosses over the very interactions that are central to the human/machine teaming experience.

In contrast, our experience developing a test harness for interactive human/machine teaming has led us to propose some additional properties that should be assessed. These properties include:

Timing sensitivity: How sensitive to exact interaction timing are results?
Sequence sensitivity: How sensitive to exact interaction order are results?
Input criticalities: Which interactions are essential to a result and which can be omitted/ignored?
Output inflections: Does the output vary smoothly as inputs change, or does it exhibit inflection points? Where are the inflection points?

These interaction-focused properties rest on the interactions that occur when the sequence and timing of user input are considered. In other words, they arise from treating user input as a *stream* that interacts with other input streams.

An architectural overview of our stream analytics test harness is shown in Fig. 2. The test harness wraps an analytic system, recording user inputs with a vector time stamp [30] derived from other input streams. The user inputs are the manifestation of interaction points in the combined human/machine teaming system. The recording is timed relative to the other input streams to enable greater flexibility in the testing. (More details can be found in a prior publication [22].) With this architecture, modeling and simulation-based tests for timing and sequence can be conducted with user inputs as the seeds of the test. Figure 3 illustrates a few such tests. Each test entails structured change of recorded input. The change may affect sequence, timing, exact values, presence/absence, or a combination of these things. By presenting slightly changed inputs and comparing the results to the original results, velocity-relevant system properties can be efficient explored. This also provides a way for user inputs to be incorporated into the software development cycle in a meaningful way, without requiring constant user input or insisting on unnatural user consistency of action.

7 Conclusion

Human/machine teams are of growing importance to data analysis. Effectively forging a human/machine team requires attention to the capabilities and limitations of each. This is especially evident for streaming data analysis because the data velocity is experienced substantially differently for the human versus the machine. However, there is much work to build on to start constructing effective human/machine team interfaces. Focusing on the interactions instead of the state kept in either side of the team provides a useful perspective on this problem.

Acknowledgments. This effort was sponsored by the Analysis in Motion Initiative at the Pacific Northwest National Laboratory. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

References

1. Baddeley, A.: Working memory: looking back and looking forward. Nat. Rev. Neurosci. **4**(10), 829–839 (2003)
2. Cowan, N.: The magical mystery four: how is working memory capacity limited, and why? Curr. Dir. Psychol. Sci. **19**(1), 51–57 (2010)
3. Miller, G.A.: The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychol. Rev. **63**(2), 81–97 (1956)
4. Cowan, N., Elliott, E.M., Saults, J.S., Morey, C.C., Mattox, S., Hismjatullina, A., Conway, A.R.: On the capacity of attention: its estimation and its role in working memory and cognitive aptitudes. Cogn. Psychol. **51**(1), 42–100 (2005)
5. Posner, M.I., Snyder, C.R., Davidson, B.J.: Attention and the detection of signals. J. Exp. Psychol.: Gen. **109**(2), 160–174 (1980)
6. Wickens, C.D.: The structure of attentional resources. Attention Perform. VIII **8**, 239–257 (1980)
7. Blaha, L.M.: An examination of task demands on the elicited processing capacity. In: Little, D.R., Altieri, N., Fifić, M., Yang, C.T. (eds.) Systems Factorial Technology: A theory driven methodology for the identification of perceptual and cognitive mechanisms. Academic Press, London (2017)

8. Heathcote, A., Coleman, J.R., Eidels, A., Watson, J.M., Houpt, J., Strayer, D.L.: Working memorys workload capacity. Mem. Cogn. **43**(7), 973–989 (2015)
9. Gluck, K.A., Gunzelmann, G.: Computational process modeling and cognitive stressors: background and prospects for application in cognitive engineering. In: Lee, J.D., Kirk, A. (eds.) The Oxford Handbook of Cognitive Engineering, pp. 424–432. Oxford University Press, Oxford (2013)
10. Su, H., Deng, J., Fei-Fei, L.: Crowdsourcing Annotations for Visual Object Detection. In: AAAI Human Computation Workshop (2012)
11. Gartner Group: Pattern-based strategy: Getting value from big data (2011)
12. McAfee, A., Brynjolfsson, E., Davenport, T.H., et al.: Big data: the management revolution. Harvard Bus. Rev. **90**(10), 60–68 (2012)
13. Newell, A.: Unified Theories of Cognition. Harvard University Press, Cambridge (1994)
14. Dasgupta, A., Arendt, D.L., Franklin, L.R., Wong, P.C., Cook, K.A.: Human factors in streaming data analysis: Challenges and opportunities for information visualization. Computer Graphics Forum **in publication** (2017)
15. Amershi, S., Cakmak, M., Knox, W.B., Kulesza, T.: Power to the people: the role of humans in interactive machine learning. AI Mag. **35**(4), 105–120 (2014)
16. Jasper, R.J., Blaha, L.M.: Interface metaphors for interactive machine learning. In: Proceedings of Human-Computer Interaction International: Augmented Cognition 2017, Vancouver, Canada, July 2017
17. Luce, R.D.: Response Times: Their Role in Inferring Elementary Mental Organization, vol. 8. Oxford University Press, Oxford (1986)
18. Townsend, J.T., Ashby, F.G.: Stochastic Modeling of Elementary Psychological Processes. CUP Archive, Cambridge (1983)
19. Amershi, S., Fogarty, J., Weld, D.: Regroup: Interactive machine learning for on-demand group creation in social networks. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 21–30. ACM (2012)
20. Arendt, D., Komurlu, C., Blaha, L.M.: CHISSL: A human-machine collaboration space for unsupervised learning. In: Proceedings of Human-Computer Interaction International: Augmented Cognition 2017, Vancouver, Canada (July 2017)
21. Endert, A., Fiaux, P., North, C.: Semantic interaction for sensemaking: Inferring analytical reasoning for model steering. IEEE Transactions on Visualization and Computer Graphics **18**(12), 2879–2888 (2012)
22. Cottam, J.A., Blaha, L., Zarzhitsky, D., Thomas, M., Skomski, E.: Crossing the streams: Fuzz testing with user input. In: 2017 IEEE International Conference on Big Data (Big Data). (Dec 2017) 4362–4371
23. Horvitz, E.: Principles of mixed-initiative user interfaces. In: Proceedings of the SIGCHI conference on Human Factors in Computing Systems, ACM (1999) 159–166
24. Cook, K., Cramer, N., Israel, D., Wolverton, M., Bruce, J., Burtner, R., Endert, A.: Mixed-initiative visual analytics using task-driven recommendations. In: 2015 IEEE Conference on Visual Analytics Science and Technology (VAST). (Oct 2015) 9–16
25. Felger, W., Schröder, F.: The visualization input pipeline-enabling semantic interaction in scientific visualization. In: Computer Graphics Forum. Volume 11, Wiley Online Library (1992) 139–151
26. Gotz, D., Zhou, M.X.: Characterizing users' visual analytic activity for insight provenance. Information Visualization **8**(1), 42–55 (2009)
27. Salvucci, D.D., Taatgen, N.A.: The multitasking mind. Oxford University Press (2010)

28. Watson, J.M., Strayer, D.L.: Supertaskers: profiles in extraordinary multitasking ability. Psychonomic Bulletin & Review **17**(4), 479–485 (2010)
29. Freeman, S., Pryce, N.: Growing Object-Oriented Software, Guided by Tests. 1st edn. Addison-Wesley Professional (2009)
30. Lamport, L.: Time, clocks, and the ordering of events in a distributed system. Commun. ACM **21**(7), 558–565 (1978)