

A General Context-Aware Framework for Improved Human-System Interactions

*Stacy Lovell Pfautz, Gabriel Ganberg,
Adam Fouse, Nathan Schurr*

■ *For humans and automation to collaborate and perform tasks effectively, all participants need access to a common representation of potentially relevant situational information, or context. This article describes a general framework for building context-aware interactive intelligent systems that comprises three major functions: (1) capture human-system interactions and infer implicit context; (2) analyze and predict user intent and goals; and (3) provide effective augmentation or mitigation strategies to improve performance, such as delivering timely, personalized information and recommendations, adjusting levels of automation, or adapting visualizations. Our goal is to develop an approach that enables humans to interact more intuitively and naturally with automation that is reusable across domains by modeling context and algorithms at a higher level of abstraction. We first provide an operational definition of context and discuss challenges and opportunities for exploiting context. We then describe our current work toward a general platform that supports developing context-aware applications in a variety of domains. We then explore an example use case illustrating how our framework can facilitate personalized collaboration within an information management and decision support tool. Future work includes evaluating our framework.*

Context awareness has recently become an important research topic. This is partly driven by rapid growth in the “Internet of Things,” which the Gartner Group predicts will grow to more than 30 billion connected devices, sensors, and computers by 2020 (Middleton, Kjeldsen, and Sutton 2013). Consequently, the types of environments, tasks, and situations in which users, machines, and intelligent services will interact are increasingly diverse. Users’ preferences and their decisions may change depending on the situation. Accordingly, systems should be able to provide the right information, at the right time, in the most appropriate format for that particular user and situation. To effectively collaborate, both humans and automation (whether robots, smart homes, or personal software assistants) need access to a common representation of potentially relevant situational information, or context. Monitoring and mining human-machine interactions and reasoning over this data enables new types of adaptive and personalized applications. But what exactly is context, how is it different from other types

of data, and why does it matter? Context seems to mean different things to different people, in domains ranging from ubiquitous and pervasive computing to organizational theory. The definitions provided in the existing literature are numerous and diverse:

The surroundings associated with phenomena which help to illuminate that phenomena, typically factors associated with units of analysis above those expressly under investigation (Cappelli and Sherer 1991).

Stimuli and phenomena that surround and thus exist in the environment external to the individual, most often at a different level of analysis (Mowday and Sutton 1993).

Any information that characterizes a situation related to the interaction between humans, applications and the surrounding environment (Dey and Abowd and Salber 2001).

What does not intervene explicitly in problem solving but constrains it (Brézillon 1999).

Situational opportunities and constraints that affect the occurrence and meaning of organizational behavior as well as functional relationships between variables (Johns 2006).

Several concepts articulated above are consistent with our characterization of context. These include the ability of context to affect the occurrence and meaning of other variables, the nature of context as constraints or opportunities, and the influence of context on interactions between agents (both human and software agents) and the environment. We subscribe to a view that is broadly inclusive yet avoids the futile strategy of modeling all potentially relevant information. Instead, we prioritize variables based on the feasibility of observing them (for example, fully or partially observable, unobservable) and on their impact to decision making. This enables us to account for the complexity of and variability in how people and agents accomplish their goals. Our working definition of context is explicit and implicit (for example, inferred, latent) information about the relationships between entities, the environment, and their interactions that may affect interpretation or decisions. Thus, our computational model of context represents the types of factors that commonly influence agents: the environment, the resources, and agent state, including their tasks, goals, and interactions.

The remainder of this article addresses three main themes: (1) motivations and challenges involved in exploiting context and building effective context-aware applications; (2) a general framework that supports contextual awareness by mining, analyzing, and adapting human-machine systems; and (3) a use case that illustrates how we have applied this work to intelligent context-aware applications.

Challenges and Opportunities

Traditionally, developing context-aware systems is a time- and labor-intensive effort that results in solu-

tions that can only be narrowly applied to the specific task of interest. There are numerous challenges in deciding which elements to model and how to adapt system behavior based on that model. Doing this successfully in a way that scales has been difficult to achieve. However, evidence continues to accumulate that doing so would be worthwhile.

Both research and common experience establish that variation in context affects performance in a variety of tasks and domains (Vlaev, Chater, and Stewart 2007; Brézillon 2003; Dey and Abowd 2000; Kozlowski and Klein 2000). This supports the notion that systematically representing and leveraging context variables can enhance human-machine collaboration. We have built context-aware adaptive systems in several domains. These systems treat humans and automation as collaborators who must build and maintain a shared mental model — a shared context — to achieve common goals in a shared environment (Ganberg et al. 2011). These systems have included proactive decision support tools, immersive multimodal environments, collaborative analytic workstations, and supervisory interfaces for teams of humans and heterogeneous unmanned autonomous vehicles. Common challenges in these domains include trust (Lee and Moray 1992), uncertainty management (Cummings and Bruni 2010), impacts of implicit coordination (Parasuraman, Mouloua, and Molloy 1996), and adjustable levels of automation (Wickens et al. 2010).

We believe that future contextually aware interactive intelligent systems will overcome several problems that plague current adaptive systems: adaptive systems fail when they do not account for user's expectations; systems that do not support individual differences are only useful for a limited set of users and contexts; and systems that fail to capture the users' intent are unable to adapt usefully. In our experience, an effective context-aware application has three major functions (figure 1): to capture the context explicitly present within the system's environment and infer implicit context, such as user state, to analyze and reason about goals and intent (including latent knowledge) within the contextual situation as a whole, and, finally, to augment the environment or situation based on its understanding of the situation. We address each of these, in turn, in the next subsections.

Context Capture: Monitoring Explicit Context and Inferring Implicit Context

Capturing and mining interaction data, especially from large amounts of users, is a big data problem. Challenges and considerations include volume, velocity, variety, and veracity. Typically one of the first tasks in building context-aware systems is to select data sources that can be made available for inferring context and implicit state. Merely collecting more data without a means to determine the val-

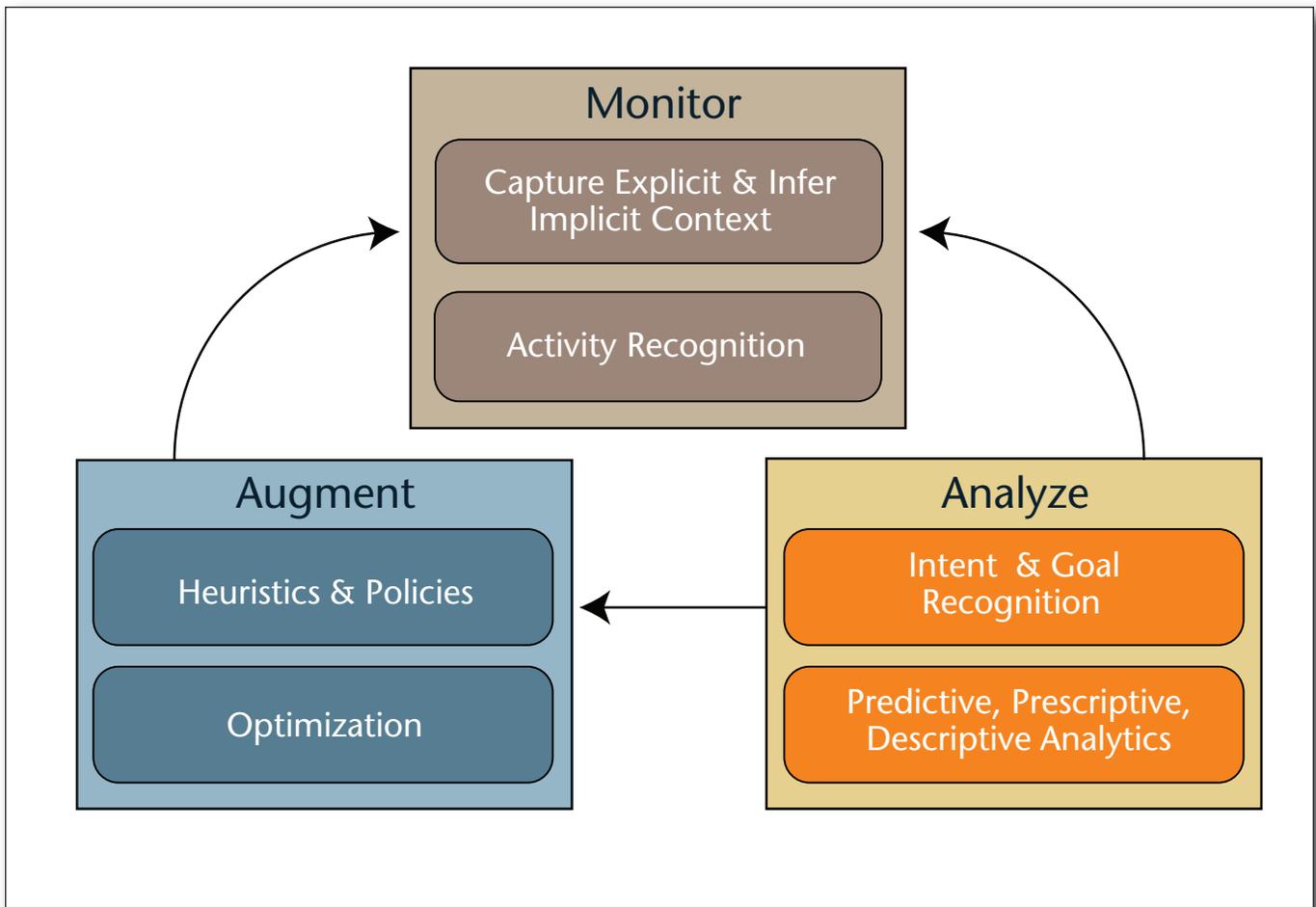


Figure 1. High-Level Model for Contextual Awareness.

ue of that data or how that data can be used to improve performance and outcomes only introduces more noise into the system with no measurable returns. Understanding cognitive and perceptual processes present in decision making can help researchers create sophisticated models for determining information value or utility (Newell 1990). Knowledge elicitation techniques have been used to identify factors that experts believe influence their attention, understanding, decisions, and actions. Because expert reports are known to be incomplete and biased (Ericsson and Simon 1993) and because such reports often identify data that are not observable (for example, prior knowledge, assumptions, hypotheses), we supplement knowledge elicitation with other methods (some of which are described below). In practice, one useful indicator that context variables are present is when researchers are questioning subject matter experts about the work domain and the experts instinctively respond “it depends” before they can provide further detail. This demonstrates a point in which detail beyond the primary task is important and the contextual variables

help to frame the decision space. Some sources of context data that we have used include system data from real world or simulated environments; sensor data from mobile devices; instrumented user interfaces that report interactions with the system; the physical state of the user based on biological or neurophysiological sensors that measure heart rate, skin temperature, and galvanic skin response; and representations of users or the environment from optical and depth cameras.

Combining traditional activity tracking with multimodal sensing enables inference about more subtle cues such as intent, goals, and user state that may be hidden to normal computer inputs. Tracking many of these signals can be accomplished through readily available commercial technologies. For example, the Microsoft Kinect uses depth-sensing technology to compute the position and orientation of a skeletal representation of a person’s body. This technology can be used to obtain position, head orientation, and even heart rate (Wu et al. 2012), which can be used to infer attention-based measures such as engagement, emotional state, and focus.

Context Analysis: Reasoning and Understanding Goals and Intent

Once the initial context variables and dimensions are defined and collected, a system must utilize one or more algorithms, models, and statistical approaches to derive meaningful insights. This includes any descriptive, predictive, and prescriptive analytics to highlight relevant trends and patterns, forecast future states, rank sets of options, or predict relevance based on interest or similarity measures. One of the greatest challenges here is finding techniques that are general or abstract enough to be reused across different types of domains and applications without losing efficiency or accuracy.

In recent years, significant progress has been made in the field of machine learning to classify, predict, and make inferences on large, complex data sets. In related work we have explored several different types of algorithms and models for representing, classifying, and inferring activities, behaviors, and tasks. In one project, we learned context and goals by observing user interactions in the multiunmanned systems control domain (Riordan et al. 2011) using Bayesian inverse reinforcement learning algorithms to efficiently learn models and context of the user with a minimal amount of domain-specific information. We have also used Latent Dirichlet Allocation, a common technique in Natural Language Processing (Yohai, Riordan and Duchon 2012), to represent events occurring in an unordered manner depending on the latent variables that define higher-level concepts. Hidden Markov models are another approach that we have used for task recognition, which allows for temporal modeling (Han et al. 2012). Such techniques can be efficiently learned to develop richer representations of context that are relatively robust to missing, conflicted, and unreliable data (Levchuk, Roberts, and Freeman 2012). Hybrid and ensemble methods (using multiple models for improved performance over any single model) are promising approaches that may address the challenges of efficiency and flexibility.

Augmentation

The third area focuses on adapting a sociotechnical system to foster collaboration between the user and the system. This is made possible by leveraging the information captured from interactions (first area), and the computational methods and analytics (second area). Optimization methods, policies, and heuristics can be used to trigger the appropriate behaviors (what action to take, when to take it, and how it should adapt) to support improved performance. Examples we have explored in past applications include selecting the right type of visualization and level of detail; providing visual, auditory, tactical, or other types of cues to users; emphasizing and filtering elements to declutter a display; providing recommendations and suggested courses of action; linking

in related/similar content; providing personalized configurations, notifications, or visualizations by learning user preferences over time; and, finally, offering to automate a task.

Augmentation is challenging because it tends to be more application and domain specific than context capture and analysis. Further, individual differences introduce added complications and variance when multitasking with automated assistance (Chen and Terrence 2009).

Toward a Generalized Context-Aware Architecture

To address these challenges, we have been iteratively refining a generalized architecture for context-aware systems over the past several years. Our initial work included mixed-initiative applications for humans collaborating with teams of autonomous vehicles. As we expanded into other domains, we separated out more generalized models and algorithms from domain-specific implementations for a more reusable architecture.

This framework, called the context engine (figure 2), is a reusable platform that serves as a run-time environment for monitoring and sharing context between applications and algorithms. It integrates with context-aware systems and provides a mechanism for running context-based algorithms. Our current implementation is written in Java and uses Titan, an open source distributed graph database, as its data store. The context engine operates in conjunction with the other systems and applications within an environment that serves as its context publishers and consumers. It uses both batch and streaming operations to include context from both large, slowly changing data sets, as well as real-time data and user interactions. For batch data ingest, it uses a series of processes to translate from static data sources to a richly linked graph model. These processes parse flat files (for example, XML, CSV), then translate the data to a property graph format. This creates entities in the system with attributes and relationships. Once the base data exists as a graph, additional processing combines multiple graph layers, infers additional relationships, and computes statistics. For streaming data ingest, it listens to incoming context change events that are produced by the various context publishers in real time, commits the changes to the graph database, triggers any necessary algorithms, and then reports the changes to any subscribing applications. It currently can receive and push information using an enterprise Java message bus as well as through HTML5 web sockets. In addition, it provides access to a RESTful web service that allows context-consuming applications to pull information by performing arbitrary graph queries.

At the core of the context engine is the common

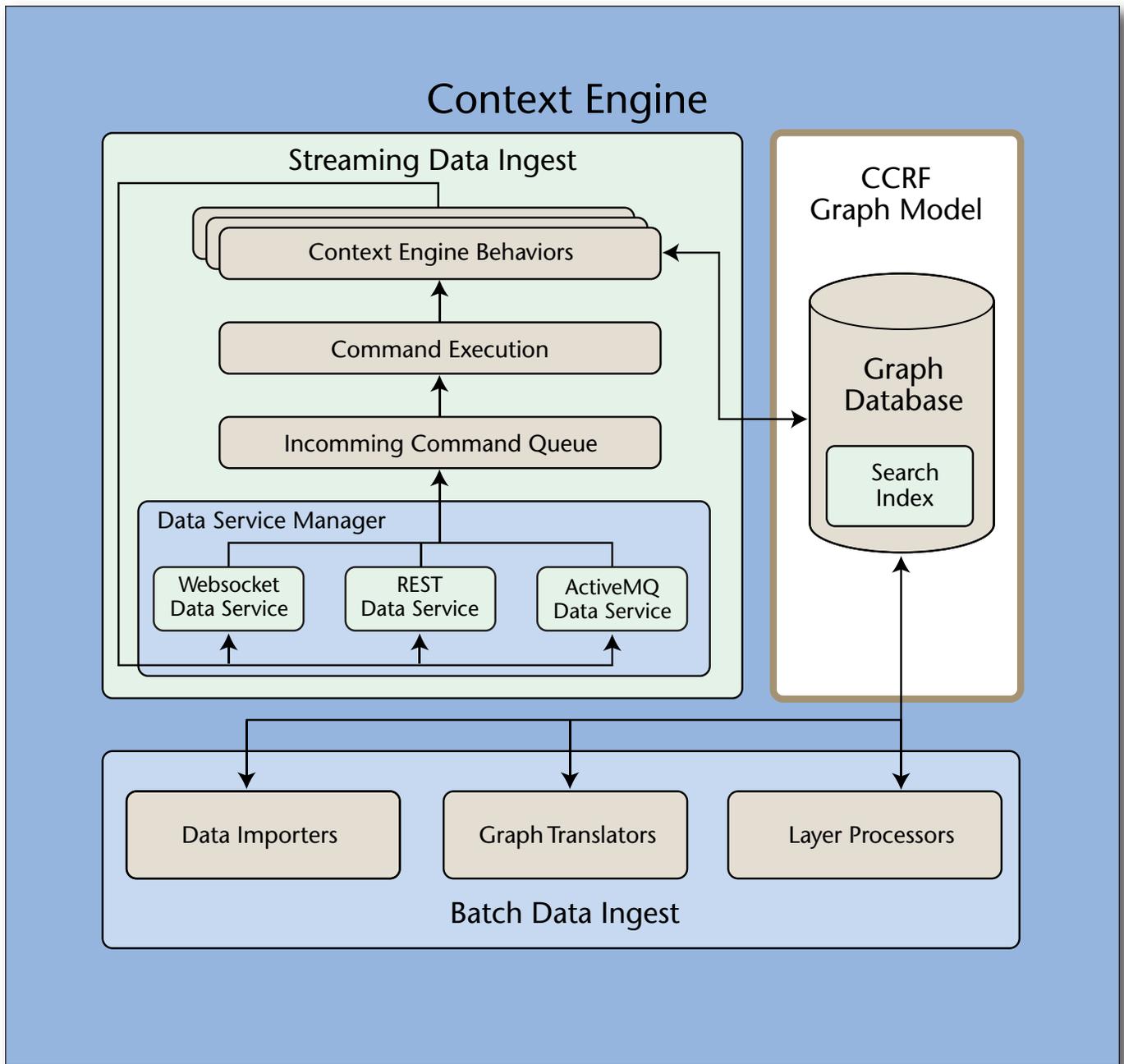


Figure 2. A General Context-Aware Framework.

context representation framework (CCRF), a conceptual model that embodies our inclusive definition of context in an abstract form. Our approach is instantiated for a particular application by creating a specific model of context that captures the key concepts and relationships for a domain. CCRF represents information as a temporal graph made of entities that have a type, a set of attributes, and a set of typed relationships with other entities. Entity attributes and relationships are tracked over time, so at any moment our picture of context consists of the cur-

rent state of entities and relationships, as well as all historical values. In addition to entities, CCRF has a concept of entity types, which define the set of attributes and relationships an entity can have. Entity types provide a mechanism for creating a metamodel that defines the key context variables for a specific application domain.

The combination of a metamodel, a set of initial entities and relationships, and a set of context inference/reasoning/action algorithms composes a CCRF domain model for a specific application (figure 3).

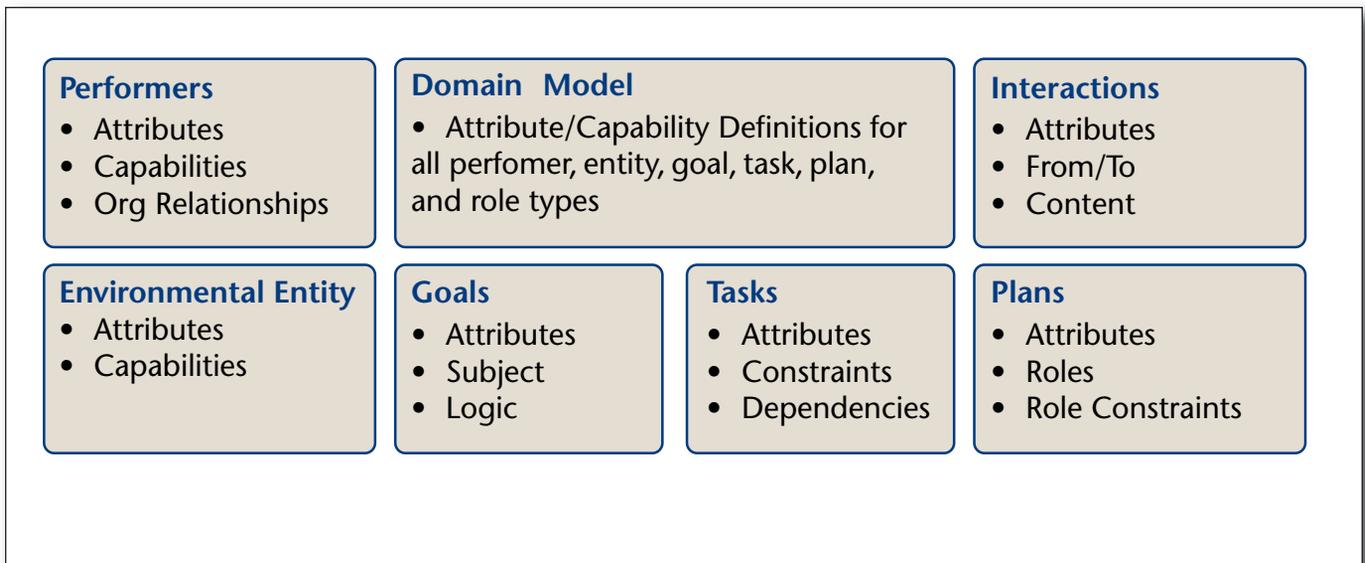


Figure 3. CCRF Domain Model for Context for Human Automation Teams.

(Ganberg et al., 2011)

Model algorithms can be very specific solutions to domain-specific problems, or they can be applicable within multiple domains, such as in the case of constraint optimization or planning algorithms. When a model algorithm is applicable in multiple domains, it should be implemented as a reusable library, and the CCRF domain model just needs to activate the algorithm library and provide a mapping from its context variables to the inputs or outputs of the algorithm. In order to instantiate a CCRF domain model, the context engine provides several Java APIs for importing/authoring metamodels and entities as well as for integrating context based algorithms.

Example Context-Aware Application

Deploying systems capable of context awareness is not an end in itself. The objective is to create highly effective and adaptive human-automation collaborations. In one application, we are applying our context framework to support personalized search and data management. Our goal is to improve the efficiency and efficacy of macrocognitive sense-making activities through a combination of automation for adapting the workspace based on contextual and cognitive factors, and immersion within the information landscape using multimodal, naturalistic human-machine interactions. We developed a use case around a team of analysts tasked with researching next-generation technology through the collection and analysis of open source information. Our approach combines several methods to infer user intent and information needs from human-machine

interaction data and exploits these models to enable more intelligent information retrieval, search, and presentation. Examples include user behaviors such as click-through rate, mouse movement, hover time, dwell time (time spent on each page), browser button use, printing, and bookmarking; user profiles such as search history and the progression of queries — both for short-term sessions and longer-term patterns and trends — demographics, role, workload; explicit context from the search query and related interests; cognitive search strategies/information seeking behavior; and group interests and information gain

Analysts collaborate with automated entity extraction services that identify semantically meaningful information in documents such as people, locations, topics, and events — as well as metadata about the documents, such as date, source, credibility, and others. As the analyst interacts with the workspace, the system builds up a semantic knowledge graph that emerges between the elements in the data and the interaction context. By combining this data, the system develops a representation of the context of analysis and provides recommendations for new information that could aid in improving both the speed and quality of analysis (for example, by diversifying searches and personalizing results). In addition to these recommendations, the system can give real-time feedback to users on the strength of their analyses to help mitigate potential problems.

In addition to information management and decision support, we have applied our framework to immersive multimodal supervisory control interfaces, geospatial analysis, and visual analytics.

Conclusion

While there is considerable progress on the approaches currently being explored and several emerging capabilities, there are still significant technical challenges and advances needed to enable the seamless integration of humans and technology.

One major challenge is to build a library of reusable algorithms that can be applied across multiple domain models. A shared infrastructure and data representation is a necessary condition but is not sufficient for the implementation of algorithms that cross-cut multiple application types. Within CCRF it is possible to formulate the model for a particular domain in many different ways, depending on the emphasis of the system. It cannot be assumed that an algorithm developed for one domain model will function with another. One possible way to address this problem will be to define data contracts for algorithms that state the types and structure of the data inputs and outputs, and to provide a translation layer that maps variables within a domain model to the variables used within the algorithm. This would essentially allow a generalized algorithm to be bound to a specific CCRF domain model assuming all the necessary data is available and the algorithm's problem formulation makes sense for the domain. Another goal is to use machine learning to adapt the system interactions to individual differences over time. In the near term, we wish to conduct several evaluations to collect more quantitative performance data.

We have presented a framework for leveraging context to support human-machine collaboration to enhance performance. First, we provided a working definition of context that has enough flexibility to work across several domains and accommodate new data and processes. The algorithms that operate in these human or automation team systems often have use across multiple types of applications and domains. If each application develops its own representation and implementation of context from scratch, then each application will also have to implement its own algorithms. Having a common application or domain-agnostic representation of context will allow application developers to focus on defining the important concepts needed for their application rather than develop a custom software implementation. They will also have access to a library of algorithms that have been developed to work with this common representation. Working from this definition, we presented a framework for representing, modeling, and reasoning about context. Finally, we discussed an example application where we leveraged our framework to build a context-aware system.

References

- Brézillon, P. 1999. Context in Problem Solving: A Survey. *The Knowledge Engineering Review* 14(1): 1–34. dx.doi.org/10.1017/S026988999141018
- Cappelli, P., and Sherer, P.D. 1991. The Missing Role of Context in OB: The Need for a Meso-Level Approach. *Research in Organizational Behavior* volume 13, 55–110. Amsterdam, Netherlands: Elsevier Ltd.
- Chen, J. Y. C., and Terrence, P. I. 2009. Effects of Imperfect Automation and Individual Differences on Concurrent Performance of Military and Robotic Tasks in a Simulated Environment. *Ergonomics* 52(8): 907–920. dx.doi.org/10.1080/00140130802680773
- Cummings, M. L., and Bruni, S. 2010. Human-Automation Collaboration in Complex Multivariate Resource Allocation Decision Support Systems. *International Journal of Intelligent Decision Technologies* 4(2): 101–114.
- Dey, A. K., and Abowd, G. D. 2000. Towards a Better Understanding of Context and Context-Awareness. Paper presented at the ACM Conference on Human Factors in Computing Systems Workshop on the What, Who, Where, When, and How of Context-Awareness. The Hague, Netherlands, 1–6 April.
- Dey, A. K., Abowd, G. D., and Salber, D. 2001. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction* 16(2-4): 97-166. dx.doi.org/10.1207/S15327051HCI16234_02
- Ericsson, K. A., and Simon, H. A. 1993. *Protocol Analysis: Verbal Reports as Data*. Cambridge, MA: The MIT Press.
- Ganberg, G.; Ayers, J.; Schurr, N.; Therrien, M.; and Rousseau, J. 2011. Representing Context Using the Context for Human and Automation Teams Model. In *Activity Context Representation: Techniques and Languages: Papers from the 2011 AAAI Workshop*. Technical Report WS-11-04. Palo Alto, CA: AAAI Press.
- Han, X.; Yu, F.; Levchuk, G.; Pattipati, K.; and Yu, F. 2012. A Probabilistic Computational Model for Identifying Organizational Structures from Uncertain Activity Data. *Journal of Advances in Information Fusion* 7(1): 78–96.
- Johns, G. 2006. The Essential Impact of Context on Organizational Behaviour. *Academy of Management Review* 31(2): 386–408. dx.doi.org/10.5465/AMR.2006.20208687
- Kozlowski, S. W. J., and Klein, K. J. 2000. A Multi-Level Approach to Theory and Research in Organizations: Contextual, Temporal, and Emergent Processes. In *Multi-Level Theory, Research, and Methods in Organizations: Foundations, Extensions, and New Directions*, ed. K. J. Klein, 3–90. San Francisco: Jossey-Bass.
- Lee J. D., and Moray, N. 1992. Trust, Control Strategies and Allocation of Function in Human-Machine Systems. *Ergonomics* 35(10): 1243–1270. dx.doi.org/10.1080/00140139208967392
- Levchuk, G.; Roberts, J.; and Freeman, J. 2012. Learning and Detecting Patterns in Multi-Attributed Network Data. In *Social Networks and Social Contagion: Papers from the 2012 AAAI Fall Symposium*. Technical Report FS-12-08. Palo Alto, CA: AAAI Press.
- Middleton, P.; Kjeldsen, P.; and Tully, J. 2013. Forecast: The Internet of Things, Worldwide, 2013. Gartner Group Technical Report (ID: G00259115). Stamford, CT: Gartner, Inc.
- Mowday, R. T., and Sutton, R. I. 1993. Organizational Behavior: Linking Individuals and Groups to Organizational Contexts. *Annual Review of Psychology* volume 44, 195–229. Palo Alto, CA: Annual Reviews Inc. dx.doi.org/10.1146/annurev.ps.44.020193.001211

Newell, A. 1990. *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press

Parasuraman, R.; Mouloua, M.; and Molloy, R. 1996. Effects of Adaptive Task Allocation on Monitoring of Automated Systems. *Human Factors* 38(4): 665–679. dx.doi.org/10.1518/001872096778827279

Riordan, B.; Bruni, S.; Schurr, N.; Freeman, J.; Ganberg, G.; Cooke, N.; and Rima, N. 2011. Inferring User Intent with Bayesian Inverse Planning: Making Sense of Multi-UAS Mission Management. Paper presented at the 20th Behavior Representation in Modeling and Simulation Conference (BRIMS), Sundance, Utah, 21–24 March.

Vlaev, I.; Chater, N.; and Stewart, N. 2007. Relativistic Financial Decisions: Context Effects on Retirement Saving and Investment Risk Preferences. *Judgment and Decision Making* 2(5): 292–311.

Wickens, C.; Li, H.; Santamaria, A.; Sebok, A.; and Sarter, N. 2010. Stages and Levels of Automation: An Integrated Meta-Analysis. Paper presented at the 54th Annual Meeting of the Human Factors and Ergonomics Society, Santa Monica, CA, 27 September –1 October.

Wu, H.; Rubinstein, M.; Shih, E.; Guttag, J.; Durand, F.; and Freeman, W. 2012. Eulerian Video Magnification for Revealing Subtle Changes in the World. *ACM Transactions on Graphics* 31(4): Article 65.

Yohai, I.; Riordan, B.; and Duchon, A. 2012. Discovering Entity Characteristics and Relationships Through Topic Modeling. Paper presented at the 4th International Conference on Applied Human Factors and Ergonomics, San Francisco, July 21–25.

Stacy Lovell Pfautz is a principal research engineer and director of the Analytics, Modeling, and Simulation division at Aptima, where she designs and develops methods for the seamless integration of humans and technology to enhance the performance of sociotechnical systems. She has served as principal investigator for several projects that combine analytics, visualization, and human-machine interaction. Pfautz received an M.S. in computer systems engineering from Northeastern University and a B.S. in computer science from the State University of New York, Plattsburgh.

Gabriel Ganberg is a principal software engineer at Aptima and technical lead for interactive intelligent systems where he builds scalable systems for ingesting, processing, querying, and visualizing complex structured and unstructured data. His interests include graph databases, distributed streaming architectures, domain-specific languages, and applied artificial intelligence. Ganberg received a B.A. degree in computer science and economics from Vassar College.

Adam Fouse is a scientist and leads the Interactive Intelligent Systems team at Aptima where he designs and develops techniques for interactive visualization of complex information. Fouse has applied an interdisciplinary approach to this research to support intelligence analysis through multimodal interfaces and to enable analysis of multiple streams of temporal data. He holds a Ph.D. and M.S. in cognitive science from the University of California, San Diego, and a B.A. in cognitive science and computer science from Brown University.

Nathan Schurr is a principal scientist at Aptima, where he focuses on the design of systems that intelligently support

AAAI Members: Watch for Your Executive Council Ballot!

Please watch your mailbox for an announcement of the 2015 Executive Council election. Ballots will be mailed in June, and will be due in mid-July. Four new councilors will be elected to serve three-year terms on the AAAI Executive Council. The Nominating Committee encourages all regular members to vote!

ICPRAM 2016

5th International Conference on
Pattern Recognition Applications and Methods

Rome, Italy · 24 - 26 February, 2016

REGULAR PAPER SUBMISSION: SEPTEMBER 10, 2015

CONFERENCE AREAS: Theory and Methods | Applications



MORE INFORMATION AT: WWW.ICPRAM.ORG

humans in complex tasks of analysis, planning, and mission execution. He has also worked on projects in domains ranging from software personal assistants to human-multirobot teams. Schurr holds a Ph.D. in computer science from the University of Southern California and a B.S. in computer engineering from California Polytechnic State University San Luis Obispo.