

Value-of-Information based Middleware for Autonomic Querying of Distributed Sensor Databases

Sreenivas R. Sukumar and Mallikarjun Shankar
Oak Ridge National Laboratory, 1 Bethel Valley Road, Oak Ridge, TN, 37830.
Email: {ssrangan@ieee.org, shankarm@ornl.gov}

ABSTRACT

With particular focus on distributed situational awareness in defense and security applications, we propose a *value-of-information* based middleware framework as a paradigm shift from crawl-index based centralized search. The proposed approach seeks to improve efficiency in search and retrieval by transforming sensors into programmable information points and enabling ubiquitous data and software flows through the infrastructure of the distributed network. We explain the different components of such a middleware framework to organize, tag and notify emerging spatial, temporal and causal patterns from the sensor measurements. We conclude the paper with a brief discussion on the top-down programming model that can realize the framework as a reconfigurable sensor query system.

Keywords: sensor query systems, search and retrieval, distributed querying, distributed data analysis

1. INTRODUCTION

The electronic revolution over the last three decades has enabled the miniaturization and cost reduction of sensor technology leading to the design, development and deployment of sensors in a wide range of applications. Today, round-the-clock operation of these sensors provides a situational-awareness security blanket protecting personnel, property and assets both inside and outside the country. However the challenge that has come along with the tremendous progress is that there are too many data sources streaming large volumes of data, forcing us to rethink and reformulate methodologies to sift through the sensor data in an efficient and reliable fashion.

As an example, let us consider a war scenario of interest, where a battlefield commander (or intelligence analyst) receives in excess of thousands of reports per hour (or on the order of dozens per second). Reports could specify suspicious movements, sniper activity, aerial attack, ground-attack with bombs, radioactivity etc. that the commander interprets and acts accordingly. In real-world situations, it is usually a small subset of the reports that hold the key to victory, while most other reports waste time and delay a critical decision in marshalling the troops. Within such unmanageable flows of information that Stew Magnuson [1] appropriately calls out as "*swimming in sensors and drowning in data*", our research is motivated by the necessity for a commander or analyst to quickly exploit incoming and archived sensor information to strategize action.

The data deluge problem transcends beyond battlefield scenarios and the Department of Defense. A similar situation pertinent to the Department of Energy is the situational awareness of the electric grid. With the recent thrust towards installing smart meters and disturbance monitoring systems within the electricity distribution and transmission infrastructure, the number of data streams that report the local operational status of the electric grid is expected to have an explosive growth. The solution for intelligently interpreting such data streams and feeding the inference back to the power-utility companies is critical in sustaining the reliability of our electric grid. Mining for power usage and outage patterns, understanding the emerging threat of cascading failures based on the spatial and temporal knowledge of status reports from distributed phasor measurement units within a short reaction time-period shares the verisimilitude of computational challenges with the battlefield sensor-data.

In this paper, we identify the need for smarter information search and retrieval techniques for large scale distributed sensor systems. We emphasize the need for autonomy in query design and content tagging over existing techniques implemented in the search engines like Google, Yahoo etc. We explain the challenges and objectives specific to distributed sensor systems and detail why existing computational techniques of search and retrieval are not architecturally best suited for sensor data streams. With emphasis on accelerating the velocity of decision making from sensor measurements, we propose a new architecture that poses sensors as programmable information points working in tandem with an application specific value-of-information interpretation middleware.

2. BACKGROUND

The state-of-the-art information search and retrieval techniques are implemented into the search engines like Google, Yahoo, Bing etc. that we use on a daily basis. If one considered the world wide web as a distributed content generation mechanism, the search engines are the query interface to the constantly evolving content. In this section, we summarize the technology behind these search engines and contrast the specific assumptions about the world wide web that makes the extension to sensor query systems non-trivial.

2.1 Crawl-Index querying

A typical search engine has three main components - (a) a crawler that finds and fetches content (from linked web pages) (2) an indexer that sorts every word on every page/document and stores the resulting index of words and occurrence in a database and (3) a query processor that matches the search query to the index to recommend relevant documents archived in the index database. The crawler, is a cluster of computers, that requests and fetches the content of documents (For example, web pages on the internet) from a host (web server) and hands over the content to the indexer. The crawler also culls out links/references from previously processed documents for future crawls and schedules crawls on important web pages based on their expected frequency of change. The crawled pages are saved in the database in a more organized form providing a query-able index structure to the text content. The index is sorted alphabetically by potential search terms in the document, with each index entry storing a list of documents in which the term appears and the location within the text document where the search term occurs. The query processor is the interface between the end user (usually the decision maker) and the indexed database implementing relevance contexts and relevance ranking algorithms to match the query interests from the end user/decision maker to the indices stored in the database. The information flow between these components within a near-exhaustive crawl-index-search architecture is illustrated in Figure 1 below.

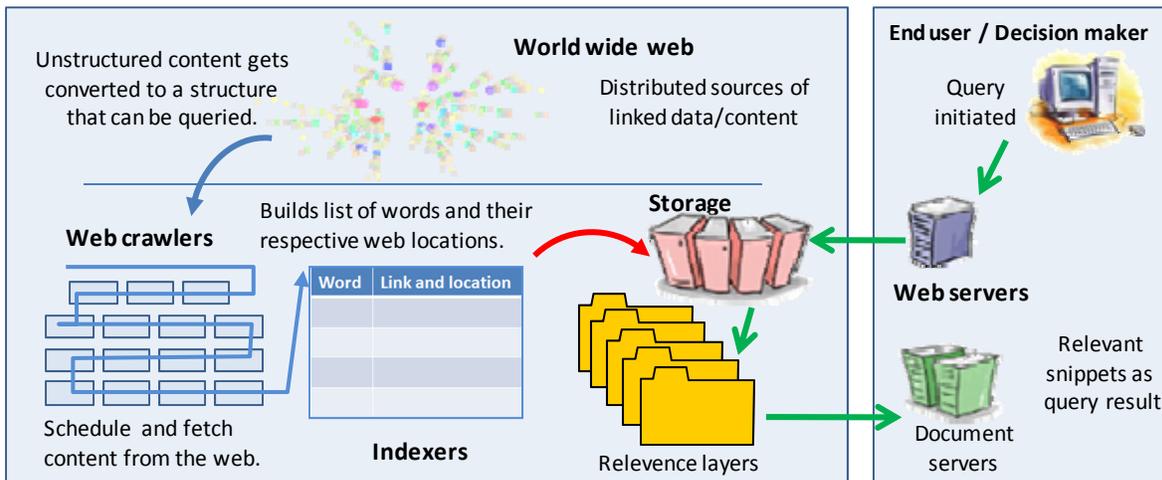


Figure 1: The search engine approach to information extraction and retrieval is built around a centralized index that is updated periodically. The accuracy and timeliness of results to a query depends on the frequency of crawling and indexing. In other words, if the significant change in content is not indexed within a short notice, query results will be obsolete and unable to capture the observed change critical to the decision maker [2].

2.2 Querying sensor data

Querying a sensor database has specific challenges to which relevance-based text search using exhaustive indices cannot be an efficient solution. A sensor database could have several modalities of streaming measurements - not only text. For example, let us consider the network of sensors in weighing stations distributed along the inter-state highways. networked cameras track trucks and record their license plate numbers. Buried sensors measure physical variables like radioactivity, weight of the truck etc. and an officer in-charge visually inspects for contraband and reports his findings. A search query could involve one or more modalities over different time periods. As number of modalities increase and the number of deployed sensors increase, the time required for indexing the data (for later querying) can be prohibitively exorbitant. The need of the hour is that decision makers should be able to forward analyze incoming sensor data with specific interests in mind, than search for it when it may be too late. That is, instead of a typical query for an index-based system that reads "Find the truck with a license plate XYZ1234", a decision maker would prefer "Notify when truck XYZ1234 arrives in Station A before 11 AM and weighs more than 3 tons."

Secondly, with distributed sensor databases one can assume a notion of proximity and neighborhood. A truck that leaves weigh station A, will have to reappear in another weigh station in the vicinity of station A within a specific period of time. In contrast, the sense of neighborhood for keywords and their location across web links is not as constrained. Thirdly, query interests (not query terms) and relevance context for actionable intelligence from sensor databases vary based on the situation. The interest in overweight trucks could easily change in a few minutes to tracking trucks of a particular color. This means that a sensor query engine has to deal with more than the "dictionary" that text-based search engines of today are programmed to index. Finally, changes in sensor measurements are unpredictable, making a scheduling approach to revisiting sensors vulnerable to costly lapses. As an example, if one scheduled crawling of transmission line status every 5 hours, an accidental outage could go unnoticed before a cascade of other failures has occurred. A notification driven forward analysis system would have alerted and prevented the cascade. Furthermore, our experience with sensor databases Sensor Net [3] and VERDE [4] has taught us that detection of emerging patterns in distributed sensor measurements are as significant as the relevance, accuracy and timeliness of query execution.

These examples, support our call for a paradigm shift from crawl-index-search architectures while querying sensor data posing the following questions. Should sensing and computing be separated in the information processing and retrieval pipeline? Why not send 50KB of software to 1GB of sensor data, rather than waiting for the 1GB of data to be indexed in a centralized server? We know that centralization of sensor data also places the burden on the reliability of the network for information flow. Can we reduce this burden by enabling ubiquitous information flows within the network of sensors? Given a search specification (i.e., an event or data of interest) and a set of data descriptors, how should we design a query system that can automatically instantiate a real-time data network to collect, organize, semantically-tag and retrieve relevant information?

3. PROPOSED APPROACH

Our proposed approach builds upon the state-of-the-art research and development in event processing to develop a novel capability that integrates ideas from complex event-processing with distributed sensing using software agents. We present the functional flow diagram of our proposed system in Figure 2 and note the participation of sensing infrastructure in the information flow in contrast to the centralized search for content indexing illustrated in Figure 1.

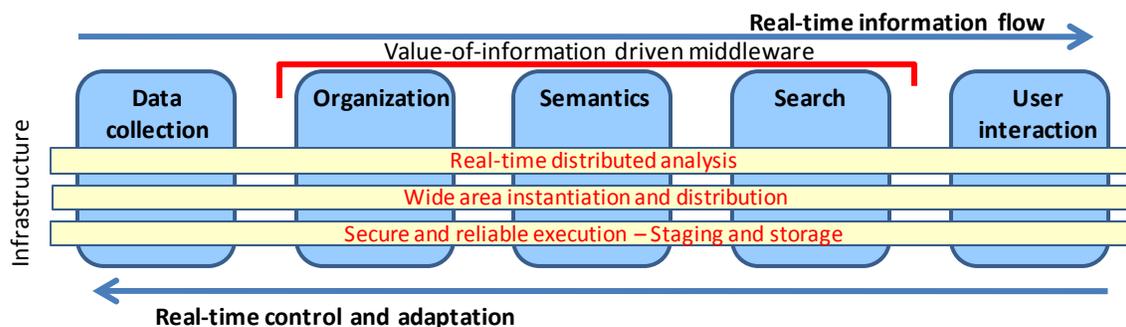


Figure 2: A component representation of our proposed approach enabling real-time information flow and control for distributed information search and retrieval.

The functional flow in our proposed architecture for distributed dynamic data search and retrieval consists of (i) data collection using sensors (ii) organization (iii) semantics - ascribing meaning (iv) search and (v) end-user/human interaction. The data collection components and the user interaction components do not differ much from the search architecture explained in Figure 1. However, in our approach, computation is executed via software agents and not a centralized server. An agent is a computer program that can move through the heterogeneous network autonomously, as needed from host to host, and can interact with other agents to solve a specific problem. Agent technologies have been evolving as a paradigm to create software that can mimic human behaviors of collaboration, autonomy, and distributed efficiency. These behaviors are desirable in solving the problem of distributed dynamic data analysis.

There are many proposed and deployed agent architectures [e.g., 5-6] in the literature. One such architecture is ORMAC (Oak Ridge Mobile Agent Community) framework [7-8] pioneered by the Oak Ridge National Lab that is widely used in the military and intelligence communities. ORMAC provides a solid foundation upon which we are able to build large-scale multi-agent systems capable of performing sophisticated tasks. ORMAC has shown demonstrable advances in distributed processing of textual and image information. We treat the successful processing of textual information as a significant breakthrough for the next logical step to distribute this capability over a heterogeneous set of computing platforms and sensor data. The search using software agents mimics common military practices in gathering intelligence - a unit may send multiple agents out to search for a commander's critical information requirements, and agents return to the unit with concise and comprehensive information based on which a strategic decision can be made. The advantages of agent technology become more prominent in situations where it is almost impossible to transmit data reliably back to the central processing point - like a low bandwidth wireless communication channel. We claim our novelty over an agent layer like ORMAC by introducing a value-of-information middleware leveraging organization algorithms, annotation algorithms and distributed search enablers.

3.1 The architectural difference

The motivation to drive an architectural difference from existing search technologies using software agents stems from battlefield observations - especially ones like Lt. Gen Peter Chiarelli [9]. His portrayal of the decision scheme compares the U.S military as a series of large dots arranged in a pyramid, each dot connected to dots above and below allowing only unidirectional flow until one reaches the top or the bottom. In contrast, his observations are that war insurgents are dots along a line, effective at passing information very quickly and efficiently. His call for the need to use portable electronics within the U.S Army to flatten the information flow of intelligence requires a redesign of how we search, retrieve and share intelligence data to make quicker and better decisions. Our proposed approach enabling ubiquitous flows using a value-of-information middleware is a step in that direction. We illustrate the idea in Figure 3 below.

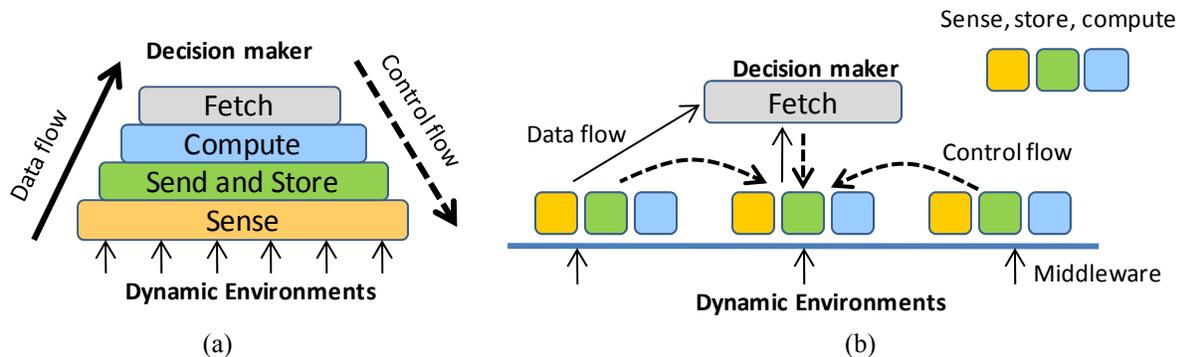


Figure 3: Architecture-level comparison of traditional search with our proposed approach. (a) Data and control flow in centralized search. (b) Data and control flow in our proposed approach flattens bottlenecks of communication and storage for retrieval purposes by enabling ubiquitous information flows. The flattened information flow architecture and the agent-based search increases the speed of fetch for the decision maker with increasing data volume compared to the traditional approach.

With the agent architecture taking care of executing software across distributed computing platforms, the challenge ahead in moving towards a distributed analysis of dynamic data streams lies in the effective implementation of the middleware - an intelligent software service component that understands contexts and optimizes data access and retrieval. The proposed middleware consists of three components of organization, relevance and programmability to direct the software agents.

3.2 Value-of-information driven middleware for search and retrieval

Without loss of generality, we will assume that sensor database measurement variables can be tabulated and organized as states of size, activity, location and time (SALT). Such a structure can accommodate reports of criminal activity in a battlefield condition or outages in the energy grid and extendable to several such applications. Our value-of-information based search and retrieval exploits this structure for organizing, tagging and annotating incoming sensor streams by assuming three roles. We illustrate the three roles in Figure 4 below. The middleware in its initial state is programmed to handle message flows from distributed sensors and archive them to an index. A new query interest gets compiled as an executable software agent that resides in the middleware that has access to both the sensors and the message flow. This software could alter what sensors detect or just the way the detection is reported in the message queue. Based on the application domain, the software agent specifies policies for organizing the messages from the sensors feeding spatial and temporal relevance. The final role of the middleware is that of supporting the existence of agents to act as "notifiers" looking out for specific patterns in the sensor streams. The middleware is also responsible for organizing and optimizing the storage of these interest patterns within the network for easy search and retrieval.

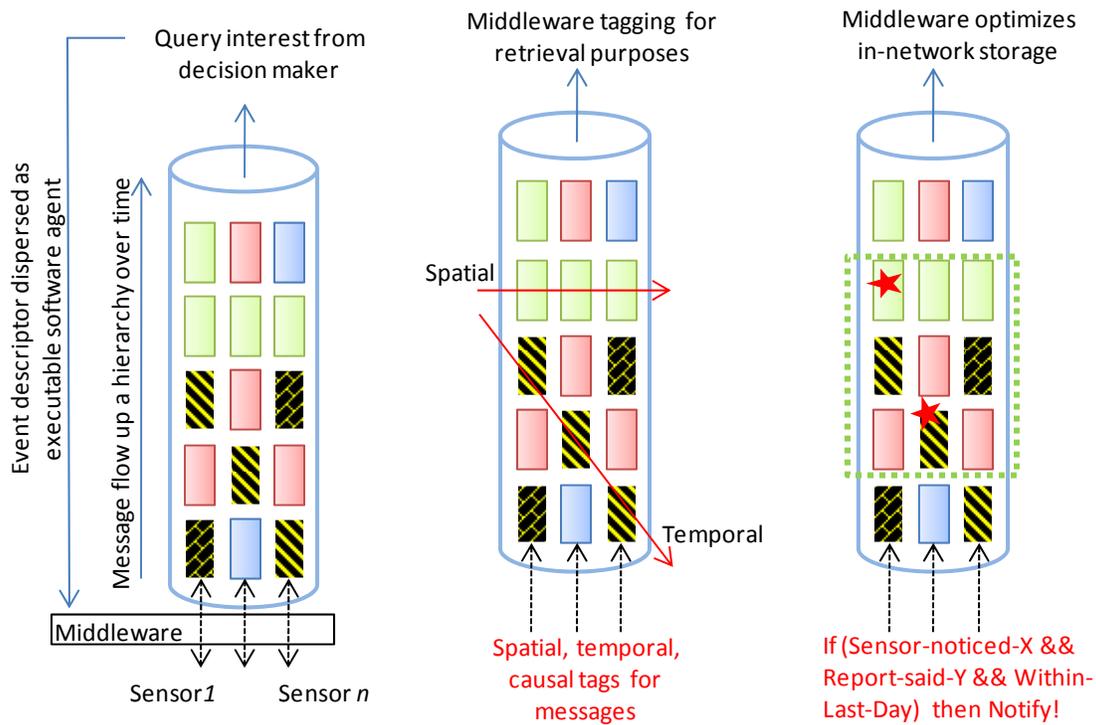


Figure 4: The three roles of the value-of-information middleware in optimizing search in distributed systems as (i) an agent router (ii) stream-tagging/organizing service (iii) a pattern detection/indexing scheme.

We get into the implementation specifications of each of the roles of organization, tagging and the compiler toolkits required for distributed actuation and instantiation in the following paragraphs.

3.2.1 Organization - A basis for distributed queries across different modalities

Figure 5 is a domain-independent generic illustration of different saliency aspects in a distributed sensor system. The saliency categorization on temporal, spatial and topical dimensions draws inspiration from previous research in models for social-organization/grouping and aggregating distributed sensory information [10]. The idea is that aggregation models and spatial clustering models can help detect emergent events from within the sensor streams if each event observed by the sensor generates a unique tag that encodes the time-stamp, space-stamp and any causal relationship with another event in the past. When a decision maker searches for similar events in future, retrieval is based on the information contained in the tag. The query does not have to sift through all of the archived data and waste time and resources instantiating the distributed network.

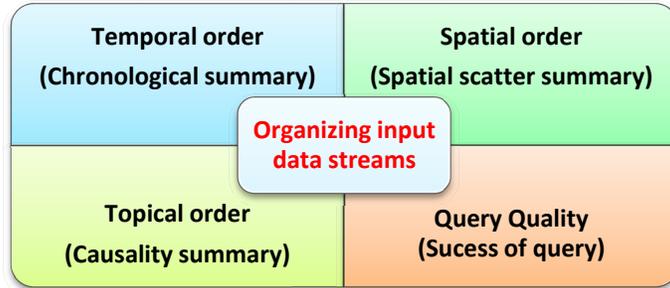


Figure 5: The middleware creates meta-data tags of the SALT streams in contexts of space, time, topic and quality.

For example, a commander may want to know whether a specified vehicle has passed through a given area in the past hour. Suppose that a network of surveillance cameras has been installed in the area, and that these cameras are constantly recording video streams. The traditional solution would be to upload the video streams of all cameras to a centralized processing facility to search for the specified vehicle, which is infeasible to do in a timely manner, more so when the search has to be done across several nodes and through a low bandwidth wireless network. Our framework would handle this problem by tasking the compute infrastructure at the nodes to search meta-data tags generated by the middleware to return a yes/no answer. The agent approach accomplishes the retrieval task by sending out 50 KB of software to the nodes instead of waiting and searching through 10 GB of the video data in the centralized repository. We note that these organization tags also enable space-time queries. A space-time query would have been really challenging with existing text-based search technologies.

3.2.2 Semantics- Defining value-of-information as an enabler for forward analysis

The next component in the value-of-information middleware is the semantic tools. The main function of the semantic tools is to ascribe meaning to streaming data. Although, we expect this aspect to be application and domain specific, we asked the question what makes streaming information valuable and tried finding answers from the literature. Different disciplines had different notions of attributing value to information. The economists [11] attach value to information if the availability of the information guides better decisions yielding higher pay-offs than the decision that one would make in the absence of that piece of information. Ecologists believe that information is valuable when the signal reliability is better than the environmental uncertainty [12]. Statisticians derive equations for information value minimizing risk on expected gains from new streams of data [13,14] building over Shannon's concepts on information theory [15]. Some argue information is valuable only when there is fluctuation between novelty and redundancy and others argue that value of information is completely contextual [16, 17]. Following the lead from computer scientists that attach value to a database by evaluating its completeness, accuracy and query ability [18] and also considering the different perspectives of research in this area, we formalize the hierarchical pyramid of information metrics illustrated in Figure 6 for distributed sensor systems.

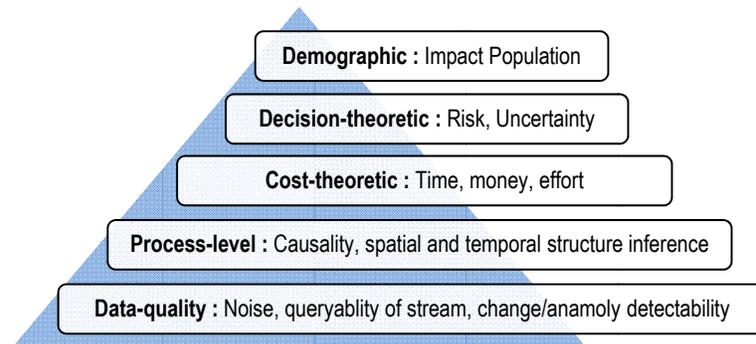


Figure 6: A generic set of value-of-information metrics for sensor query systems. These metrics act as relevance context to schedule and prioritize queries instantiating a distributed network of sensors seeking actionable intelligence.

The value-of-information metrics when implemented as executable software in the middleware are a means to estimate the expected benefits (in terms of cost and impact) of executing a query. The application domain would dictate the number of levels to include from the pyramid in Figure 6. These expected risk estimates guide the decision to prioritize one query over another autonomously.

3.2.3 Search - Distributed actuation using event processing language specification

The final component in the value-of-information middleware is the programming model for distributed actuation and execution. We propose novel extensions leveraging ideas from time-tested structured query language constructs and relatively new data stream management constructs. [19, 20] More specifically, we enable search and information extraction in a generalized form: "EXTRACT *EventDescription*(d_1, \dots, d_n, t) FROM $\{D_1, D_2, \dots, D_m, t, location\}$ WHERE *ConditionalFunctions*(\mathbf{D}, t)", where the *EventDescription* is a state-change of interest described syntactically (note that d_i are individual data descriptions and t is time), D_i are descriptors of the consolidated sources of data and *ConditionalFunctions* are generalizations similar to the where clause in Structured Query Language to allow distributed spatial and temporal constraints. A query written in this extended language maps to the distributed real-time overlay in the network through a compilation process.

Once a query is issued, we create agents of executable software that can reside at the collection nodes to forward analyze the data or sift through value-of-information meta tags created by the middleware. The query quality component in the organization middleware (Figure 5) infers when and how new stream data retires older streams while also determining the subsets of stream information that should be preserved for rapid search while archiving the raw data for later analysis.

4. SUMMARY

We have addressed a real and worsening data deluge problem in homeland security and defense applications, where decision makers and analysts must review vast amounts of information. We noted that data flowing from different sources/sensors at high rates can easily overwhelm human analysts in the military or in national security organizations. When the amount of data gets large or the network bandwidth is limited (e.g., in wireless settings), the traditional approaches of crawl-index-search breaks down. Further, the typical approach of pulling data from a central repository creates a single point of failure and generates unnecessary transmission of data through the network. Also, network traffic to a centralized data repository can be targeted by a malicious enemy.

We identified these issues and proposed a distributed top-down programming model for reconfigurable sensor query system and argue for information search efficiency by:

- Transforming sensors (devices and agents) into programmable information points for information extraction and retrieval.
- Introducing a new middleware design to enable ubiquitous flows with actuation and instantiation of distributed resources toward network-bottleneck-free operations.
- Providing the means for tasking infrastructure and enabling forward analysis of sensor data streams.

ACKNOWLEDGEMENTS

This manuscript is authored by employees of UTBattelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy. Accordingly, the United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

REFERENCES

- [1] S. Magnuson, "Military swimming in sensors and drowning in data", *National Defense Magazine - Sensor overload: U.S forces bogged down in battlefield data*, pp. 36-38, 2010.
- [2] C. Sherman and G. Price, "The invisible web: Uncovering information sources search engines cannot see", *CyberAge Books*, 2001.
- [3] B.L. Gorman, M. Shankar and C.M. Smith, "The SensorNet Node: Last-Mile Platform for Interoperability", *Sensors Magazine*, Vol. 22, April 2005.
- [4] M. Shankar, J. Stovall, A. Sorokine, B. Bhaduri, T. King, "VERDE: Visualizing Energy Resources Dynamically on Earth", in the Proc. of the Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, pp. 1-4, 2008.
- [5] K. Sycara, A. Pannu, M. Williamson, and D. Zeng, "Distributed Intelligent Agents," *IEEE Expert*, Vol. 11, No. 6, pp. 36-46, 1996.
- [6] M. Griss and G. Pour, "Accelerating Development with Agent Components," *IEEE Computer*, Vol. 34, No. 5, pp. 37-43, 2001.
- [7] J. W. Reed, T. E. Potok, and R. M. Patton, "A multi-agent system for distributed cluster analysis," in Proc. of Third International Workshop on Software Engineering for Large-Scale Multi-Agent Systems (SELMAS'04), W16L Workshop - 26th International Conference on Software Engineering, Edinburgh, Scotland, UK: IEE, pp. 152-155, 2004.
- [8] P. Palathingal, T. E. Potok, and R. M. Patton, "Agent based approach for searching, mining and managing enormous amounts of spatial image data," in the Proc. of the Eighteenth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2005 - Recent Advances in Artificial Intelligence, pp. 351-356, 2005.
- [9] W. Maththews, "Helping information flow freely- Insurgents outdo U.S Military", *Defense News Magazine*, January 2008.
- [10] F. Dressler, "Self-Organization in Sensor and Actor Networks". *Wiley & Sons*, 2007.
- [11] D. W. Hubbard, "How to measure anything - Finding the value of "intangibles in Business", *Wiley and Sons*, 2007.
- [12] C. M. McLinn and D.W. Stephens, "What makes information valuable: signal reliability and environmental uncertainty", *Animal Behavior*, Vol. 71, pp. 1119-1129, 2006.
- [13] R. Howard, "Information value theory", *IEEE Transactions on Systems Science and Cybernetics*, Vol. 2, No. 1, 1966.
- [14] M. Chu, H. Haussecker, F. Zhao, "Scalable information-driven sensor querying and routing for ad-hoc heterogeneous sensor networks", *The International Journal of High Performance Computing Application*, Vol. 16, No. 3, pp. 293-33, 2002.
- [15] C. E. Shannon, "A mathematical theory of communication", *The Bell System Technical Journal*, vol. 27, pp. 379-423, 623-656, July, Oct. 1948.
- [16] A. Fenner, "Placing value on information", *Library Philosophy and Practice*, Vol.4 , No. 2, 2002.
- [17] P. P. Pirolli and S. K. Card, "Information Foraging", *Psychological Review*, Vol. 106, No. 4, pp. 643-675, 1999.
- [18] R. Y. Wang , V. C. Storey , C. P. Firth, "A Framework for Analysis of Data Quality Research", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 7, pp. 623-640, 1995.
- [19] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and Issues in Data Stream Systems", Invited paper in Proceedings of the symposium on Principles of database systems, pp. 1-16, 2002.
- [20] V. Markl, G. M. Lohman, V. Raman, "LEO: An autonomic query optimizer for DB2", *IBM Systems Journal*, Vol. 42, No. 1, 2003.