Customer-Driven Sensor Management

Tracy Mullen, Viswanath Avasarala, David L. Hall
School of Information Sciences and Technology
Penn State University
{tmullen, vavasarala, dhall}@ist.psu.edu

Abstract

Customer-driven sensor management advocates bringing ecommerce concepts and advances to bear in sensor management. In ecommerce, customer wants essentially drive the production process. Sensor management has traditionally followed a much less capitalistic process, producing information "goods" based on pre-defined system goals and priorities. We explore here some of the possibilities of incorporating a customer-driven market-based approach to sensor management.

Keywords: *sensor management, market-based approaches, combinatorial auctions, level 4 data fusion, genetic algorithms, local stochastic search.*

Introduction

Customers are essential in e-commerce, and this is reflected in the abundance of references to customer-driven marketing or customer relationship management. However, such verbiage is not found in the sensor management field. While some might argue that this is a good thing, underlying this terminological difference is also a methodological one. The sensor management field has traditionally focused on producing and processing as much information as possible rather than on customer-driven/user-aware information processing. Such an approach effectively disconnects the user from the sensor "production" engine. In the past, sensor resources were highly constrained in how they could be used, and users were content to get whatever they could, whenever they could. New technologies, such as DARPA's smart dust program, mean that sensors are becoming smarter, smaller, and cheaper. While challenges remain in the field of sensor design and development, an emerging problem is not so much the nuts and bolts of producing new information, it is what to produce, where, for whom, and when.

E-commerce businesses such as Amazon.com provide a potent analogy for self-managing information systems that must respond to changing user goals and demands. Just as users do not know where sensors reside, customers do not know the location or ownership of Amazon's book storage sites. Their web site provides a front-end to various intermediary services that help customers specify what is available and what they are interested in. Some of these intermediary services are straightforward, such as when the customer searches for a title/author combination. Some are more sophisticated, such as using collaborative filtering to identify users with similar tastes, and recommending new books based on what these users have purchased. Similarly, one could imagine a sensor management system that recommends a new information product (e.g., a target track) based on what previous users in similar situations have selected.

In a customer-driven sensor management system, sensors and models provide data to information aggregation and fusion processes. These middlemen transform the data into requested information, with customers driving the information transformation process. Potential sensor management application areas include not only typical defense ones, but also aircraft health and usage monitoring [1], crisis management [2], and self-managing content distribution in vehicular networks [3].

New sensors and wideband communications provide an information rich environment for users, shown conceptually in Figure 1. The next generation of sensor management platforms must manage and synchronize the activities of increasingly intelligent sensors and processes across distributed platforms. The prevalence of Internet technologies means that human observers can, in some sense, be considered sensors embedded in the network, albeit with vastly different time horizons than the typical sensor. Consider network-centric warfare applications [4], where multiple sensors such as acoustic, seismic, infrared and radar provide information about potential targets, activities and events. Human analysts interpret these data to make inferences about an evolving situation or threat. Thus in such an application humans can act both as a provider of information (via personal observations) as well as a user of information.

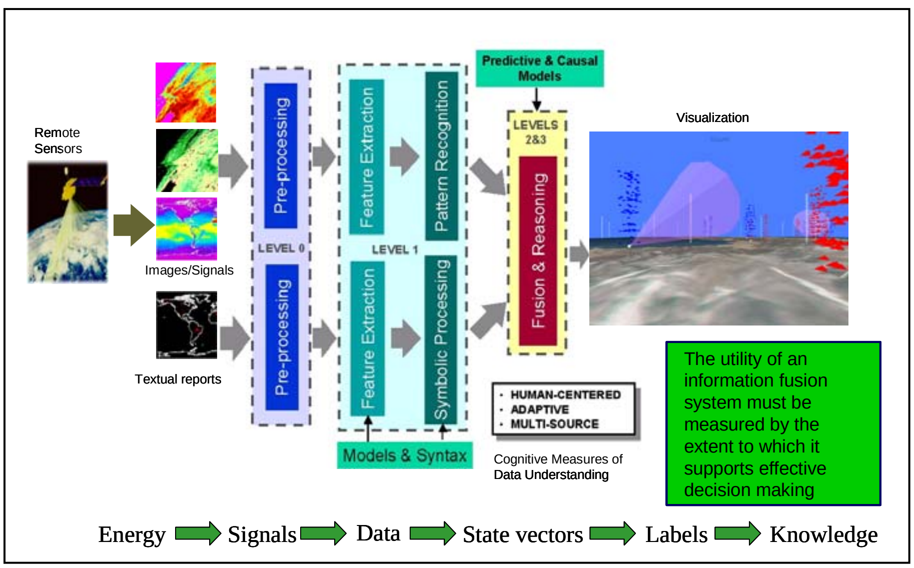


Figure 1: Energy to knowledge

The sensor node modules in sensor networks range considerably in size and capability. Small sensor node modules are often referred to as motes, and include commercial versions such as the MICA2 mote offered by Crossbow Technologies. The processors in such small motes can process four million to 10 million instructions a second, whereas the processor in a handheld computer (e.g., PDA) can perform about 400 million a second [5]. On the other end of the spectrum are higher capability sensor node modules often referred to as gateways, such as the Crossbow Stargate module powered by the Intel Xscale PXA255 processor. The Xscale processor is the same processor found in a number of high end handheld computers. The Tenet architecture [6] proposes that large-scale sensor networks be tiered. Lower-tier motes provide dense sensing, while the higher-tier can be 32-bit master nodes with more powerful communication capabilities. They note that the cost and complexity of implementing data fusion and complex algorithms in motes outweighs the performance benefits.

Our focus is on the higher-tier level, which has processing and communication constraints, but not the extreme constraints of the sensor networks composed of motes. We also focus on systems with multiple decision makers who may report to distinct, albeit cooperating, organizations. Given that these decision makers have different mission objectives, maximizing over all of them requires rapidly detecting and reacting to new situations with an optimum combination of available sensors. Further, there exists underlying uncertainty in whether and how quickly sensors can acquire needed information, and this can vary depending on the sensor resources used, observing conditions (e.g., weather, terrain, etc.), and possible actions by the observed target.

To allocate sensor resources, a sensor management system must evaluate the tradeoffs involved in deploying various types of devices (e.g., better information gain vs. greater potential for detection) and in fine-tuning sensor controls (e.g., resolution vs. update rate). It must also consider potential interactions (positive and negative) with other deployed sensors. Faced with such complex and highly integrated sensor management scenarios, traditional optimization techniques either gravitate towards centralized solutions that do not scale well, or “point solutions” that do not generalize well. So we are faced with the question of how can diverse sensing resources be used effectively to support multiple users having different priorities and time horizons? Especially when these resources may be dynamically available (e.g., coming into and out of coverage, having widely different performance based on local environmental conditions, etc.) and may even be unknown to potential users.

We believe that a market-based approach [7] provides a valuable framework to address such dynamic distributed resource allocation problems. The basic idea is that, just as in human societies, market mechanisms can facilitate decentralized resource allocation involving complex tradeoffs of goods and services in computational systems. While historically, such methods have not been considered competitive with more traditional optimization methods, two factors make market-based architectures increasing appealing: (1) the movement towards network-centric and service-based operations (including human-in-the-loop decision-makers and resource requestors), which make traditional optimization techniques less suitable, and (2) advances in electronic commerce that have increased the effectiveness and efficiency of market-based techniques. Our main contribution lies in identifying and developing the application of such e-commerce directed market-based research to sensor management. While this technique may not be appropriate for all sensor management problems, especially the extremely resource-bounded ones, we believe it has merit for a wide class of resource-constrained situations.

Sensor Management Overview

The term *sensor management* refers to “the process which seeks to manage or coordinate the use of sensing resources in a manner that improves the process of data fusion and ultimately that of perception, synergistically" [8]. The Joint Directors of Laboratories (JDL) data fusion processing model [9] considers sensor management a Level-4 process, see Figure 2. Typical Level-4 sensor management tasks include sensor scheduling and determining the significance/criticality of any task to the overall mission. While our inclusion of human-in-the-loop decision making does entail Level 5 functionality, that aspect is not currently the main thrust of our work.

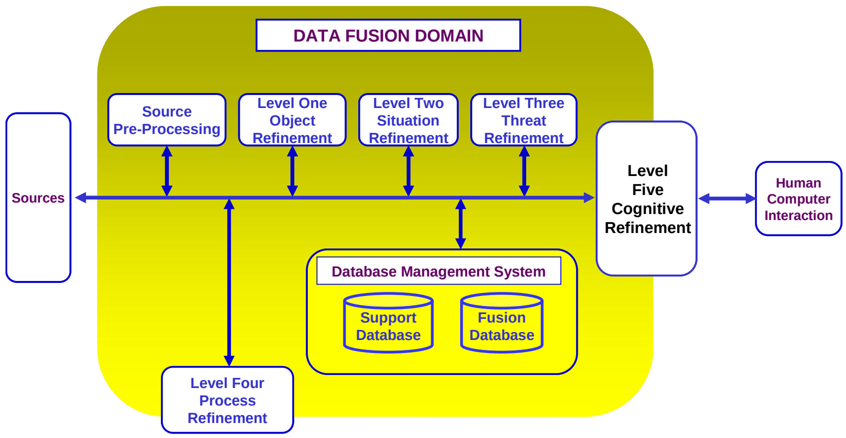


Figure 2: Joint Directors of Laboratories (JDL) Model

Sensor performance and sensor management are intimately linked—knowledge of sensor performance (e.g., resolution, fidelity, update rates) is required to improve data fusion results. Sensors provide information about the environment (e.g., environmental scans) and sensor managers aim to provide an optimum sensor configuration based on predicted system performance. This includes determining how best to deploy, configure, or reconfigure sensors, subject to processing, battery or communication constraints, so that mission objectives are met effectively.

Our literature survey, described in [10], shows that research in this domain is incomplete in several areas:

a) Most solutions are "point solutions" that do not use generic sensor management architectures, and thus their applicability beyond their original test beds is not guaranteed.
b) Lack of decentralized/distributed control mechanisms means the sensor management techniques not scale well and may impair real time performance.
c) Lack of "human in the loop" means that the sensor management architecture does not extend well to a system involving humans as information producers or consumers.
d) Neglect of "Value of information"– Traditional sensor management approaches like the information theoretic sensor management have concentrated on optimizing the overall quantity of information obtained using the sensor resources, neglecting the value of information to the overall mission goal. Only very recently have researchers started to address these issues. Dynamic goal lattices [11], decision-making theory [12] and Bayesian networks [13] have been used to find the priorities/weights of various objectives for sensor management optimization. However, a comprehensive paradigm aimed at taking mission objectives directly into consideration during

resource allocation is still lacking. Market-based approaches provide a rigorous methodology for incorporating the value of information to mission goals during resource allocation. When distinct organizations use the same sensor manager, complete information about any individual user's situation may not be available. In this case, it can be difficult for a traditional sensor manager approach to decide on the priorities associated with the different requests for resources.

Market Architecture for Sensor Management (MASM) architecture

To address the above issues, we model a sensor management system as a computational economy, and are testing it on increasingly realistic environmental models (e.g., more realistic targets and target motion, sensor models, mission timeliness constraints, etc.). In this paper, we focus on sensor management requirements of a single platform with multiple heterogeneous assets. However, we are extending this model to multiple coordinating platforms to handle systems of distributed sensing systems. To scale up our system requires distributed auction design, which has been explored in [14-16]. However, given real-time and bandwidth constraints, the bulk of resource allocations tend to be local ones.

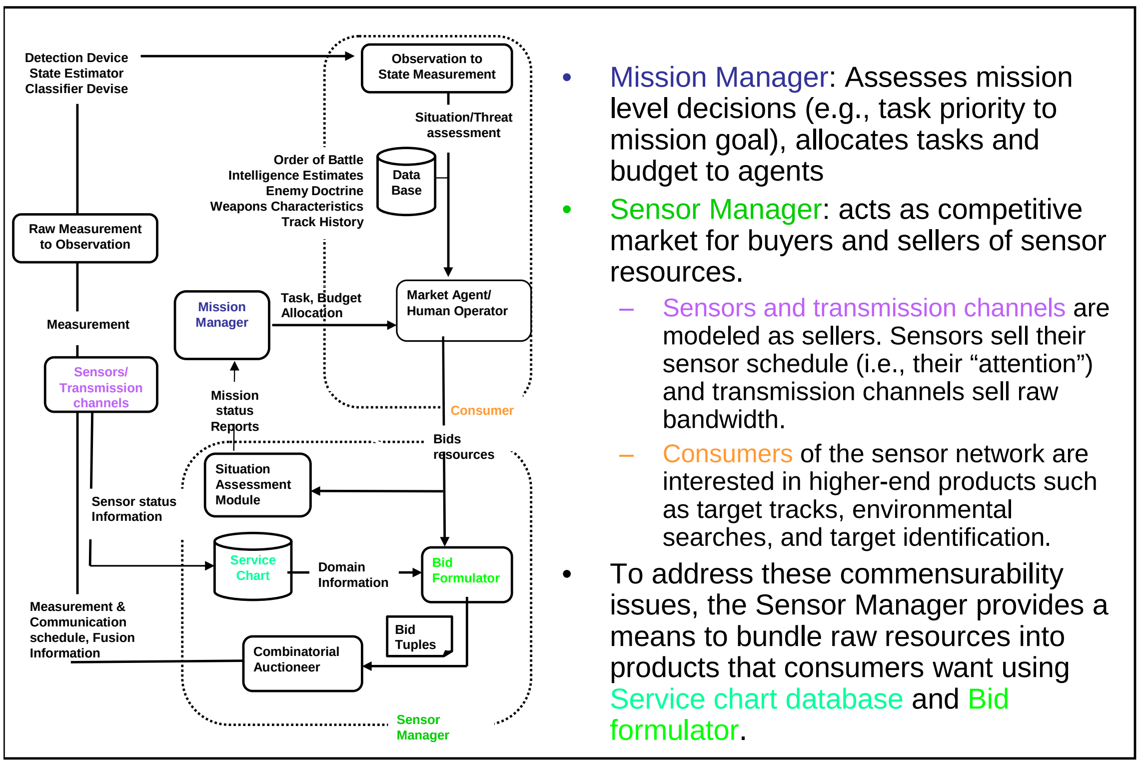


Figure 3: MASM Single Platform Architecture

Our single platform design is shown in Figure 3, and derives from the sensor management architecture proposed by Denton et al [17]. The main components are the mission manager and sensor manager. The mission manager component is responsible for assessing mission level decisions such as the priority of various tasks to mission goals, and allocating these tasks to agents. Within the mission manager, approaches such as goal lattices can be used to measure the criticality of various low level tasks to the overall mission goals, and thus help to determine their respective budgets. Hintz et al [6] have used goal lattices to compute the relative weights of actionable tasks like tracking based on high level mission goals. Muraleedharan et al. [18] have used goal lattices to determine weights for combining various objectives to optimize routing in a sensor network. Newer developments include dynamic goal lattices to support more dynamic goal generation from a set of predefined goals [11]

Table 1 shows an incomplete subset of typical USAF goals draw from an "in harm's way" scenario developed in [19]. These high-level goals are mapped to weights for the lowest-level goals which are real, measurable actions. These weights can be construed as the relative importance of the low-level actionable goals to the overall mission goal.

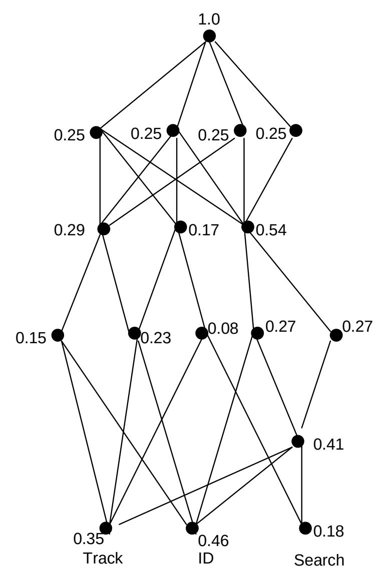
Goal	Description	Included Goals	Goal Lattice
1	to obtain and maintain air superiority	2,3,4,5	
2	to minimize losses	6,7,8	
3	to minimize personnel losses	6,7,8	
4	to minimize weapons expenditure	6,8	
5	to seize the element of surprise	8	
6	to avoid own detection	9,10	
7	to minimize fuel usage	10,11	
8	to minimize the uncertainty about	12,13	
9	to navigate	15,16	
10	to avoid threats	15,16	
11	to route plan	15,17	
12	to maintain currency of the enemy	14,16	
13	to assess state of the enemy's	14	
14	to collect intelligence	15,16,17	
15	to track all detected targets	15	
16	to identify targets	16	
17	to search for enemy targets	17	

Table 1: Subset of USAF goals

The sensor manager (SM) component is responsible for allocating sensors to various tasks, sensor scheduling and fault diagnosis. Our sensor manager allocates all necessary network resources, including communication bandwidth and battery power. This allows it to consider tradeoffs, such as battery power vs. communication, that often simply have to be hard-wired into other systems. Individual sensors take measurements of the environment, as per the schedule given the sensor manager, and communicate these measurements to a fusion center. The fusion center is generally another sensor that performs any necessary data fusion processing before sending the information back to the consuming processes. The SM selects the fusion center based on constraints such as battery power, processing capability, and communication requirements. The output of the fusion center is sent to the requesting consumer.

The sensor manager acts as a competitive market for buyers and sellers of sensor resources. Sensors and transmission channels are modeled as sellers. Sensors sell their sensor schedule (i.e., their "attention") and transmission channels sell raw bandwidth. However, consumers of the sensor network are interested in higher-end products such as target identity. In the case of traditional defense sensor management applications, these higher-end products would be target tracks, environmental searches, and target identification. To address these commensurability issues, the SM provides a means to bundle raw resources into products that consumers want using a service chart database and a specialized bid formulator module.

The service chart specifies detailed domain information, such as the location and characteristics of various sensors in the field and available communication bandwidth. The SM accepts consumer bids, which it then hands off to the bid formulator. These bids are in the form of ***<t, p>*** where *t* is a task description including the minimum task quality that is acceptable to the consumer and *p* is the price the consumer is willing to pay. Using the service chart, the bid formulator enumerates the different possible allocations for each task and uses the consumer bid price to determine price quotes for each possible allocation. As part of this process, the bid formulator needs to evaluate the value of a particular allocation to the overall consumer task. For example, it may be willing to pay more to buy a specific sensor with "excellent" applicability to a perceived threat, than to get another type of sensor with just "good" applicability. It must also consider synergist constraints, such as sensor A working in conjunction with sensor B may be able to provide more accurate tracking than the more powerful sensor C working alone.

Once the bids are formulated, the sensor manager conducts an auction to set prices and allocate the available sensors as needed. Auctions that sell bundles of goods are called *combinatorial auctions*. Buyers can express their preferences over a combination of goods, thus efficiently represent synergies (or lack of) between goods. A typical example is that most buyers would much prefer to buy the combination of a left shoe and a right shoe over getting either one independently. In a combinatorial auction, bidders offer bids of the form $B_i = \langle b_i, p_i \rangle$, where b_i is the bundle of resources and p_i is the price the bidder is willing to pay for the bundle b_i. The winner determination problem is to find an allocation of resources that maximizes revenue given the constraint that each resource can be sold to no more than one bidder. Formally, this problem can be characterized as [20]:

$$\max \sum_{j=1}^{n} p_j x_j \;\; s.t. \sum_{j|i \in S_j} x_j \le 1, \forall i \in \{1..m\}$$

where x_j is 1 if the bid is accepted to the final allocation and 0 otherwise. This problem is NP-hard and many approaches have been proposed in literature. Most of them focus on developing an exact winner determination procedure using mixed integer programming or specialized algorithms. CABOB has proven to be one of the fastest exact winner determination algorithms, and uses a specialized depth-first search on the space of possible bids [20]. However, exact winner determination may not be feasible in scenarios when there are a large number of bids (i.e., combinations of resources) to be allocated. For combinatorial auctions to be useful in environments with strict, real-time constraints, their anytime performance is critical. For this purpose, we investigated approximate winner determination algorithms and performance. We developed SGA (seeded genetic algorithm), which uses a novel representational scheme to produce only feasible allocations.

We compared the anytime performance of SGA with Casanova [21], a local stochastic search procedure. Algorithm performance varies not only with the number of bids (or sensors), but also with dynamic characteristics described by the bid distribution. The decay bid distribution is one of the toughest bid distributions for CABOB, and was used (among others) to compare SGA and Casanova performance. A problem instance for the decay distribution is described by $P(n, m, \alpha)$. In this distribution, there are n bids. For each bid, with probability α, a new item is randomly chosen without replacement and added until m items have been added. We selected the parameter settings of n = 800, m = 80 and α = 0.75, where the average running time reported for CABOB was 1000 seconds. Figure 4 shows the real time performance of SGA and Casanova averaged over 20 runs. While the results from CABOB are those reported in [20], and not run on the same machine as the SGA and Casanova tests below, we would expect the runtimes to be roughly comparable. The two algorithms have similar performance, although Casanova tends to do better under very strict real-time constraints (i.e., less than 5 CPU seconds) while SGA tends to do better when constraints are slightly relaxed. We are currently investigating the use of a seeding procedure which uses Casanova to seed the SGA.

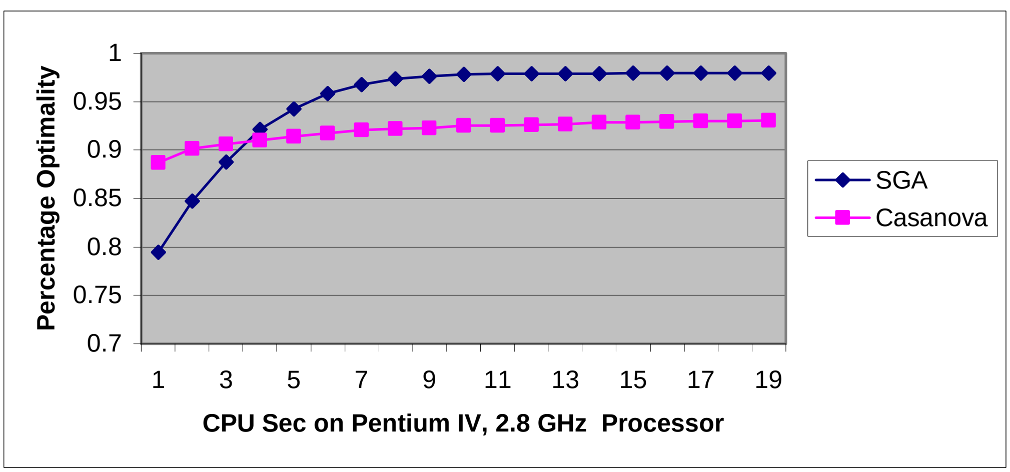


Figure 4: Comparative Real-time Performance of SGA and Casanova

Adapting to Real Time Constraints

Depending on the size of network and the real-time constraints, MASM can provide the service chart/bid formulator functionality in 2 different modes, either explicit service mappings or dynamic service mappings. If the number of sensors is small and the real-time constraints are relaxed, then MASM can provide an explicit service mapping. In this mode, the service chart database computation is done offline, delineating an explicit mapping between tasks and service providers (e.g., sensors, information sources, communication channels). It contains the percentage of utility obtained for each task that the consumers can bid on, for every combination of sensors that can be used to accomplish that task. When a consumer bids for a task, the bid value for a particular service is calculated as the product of the percentage utility achieved by the service for that particular task as shown by the service chart and the bid value. Thus, if the number of sensors in the network is n and the number of task types that can be bid on by consumers is *m*, the service chart has $2^n m$ entries. If the number of bids received by the SM is *k*, then the total number of bids in the combinatorial auction is $2^n mk$.

However, creating a data-base with $2^n m$ entries will not be feasible in case of large networks because of memory constraints. Moreover, in a large dynamic environment, the explicit mapping between tasks and service providers may not be feasible to specify. The number of bids in the CA auction problem can be very large in the case of large sensor networks. This increases the computational time required for the winner determination problem. For these scenarios, MASM provides dynamic service mappings. The database of the service chart is replaced by a function-estimation neural network. The neural network is trained offline to output the percentage of utility obtained for a given combination of sensors for different task parameters. Instead of the bid formulator explicitly formulating combinatorial bids for each consumer bid, the search space is directly searched using SGA. The representation schema used for the algorithms is similar to the one used for winner determination in combinatorial auctions and is as follows. Let the number of bids be k and the number of sensors be *n*. The representation schema is a real valued string of length n, where each string bit is a real valued number between *1..k*. The *j*-th string member essentially represents the bid to which to sensor j is allocated. The overall utility obtained for this string is obtained as the sum of utilities obtained from individual bids, which is in turn estimated using the neural network. Since function estimation in a neural network and the genetic algorithm are both polynomial in complexity and SGA is anytime, the overall allocation determination is only polynomial and anytime. Finally, the neural network can be used to learn mappings between tasks and service providers as the situation evolves.

Small Sensor Network Comparison Results

The small sensor network comparison below illustrates the complexities associated with a multi-consumer, multi-goal sensor network environment, and allows for incorporation of challenging issues in sensing and mission needs. For our initial research in this area, we designed and implemented a simple comparative study of information-theoretic and utility-based sensor resource allocation. Direct information-theoretic measures concentrate on maximizing the quantity of information, neglecting the value of information for mission goals. On the other hand, utility-based measures use the value of information for the mission goals in the sensor manager's decision-making process. To conduct this comparative study, we developed a simulation environment consisting of a two dimensional search area involving multiple targets and multiple sensors for testing and comparing the performance of the various sensor management techniques. The search area is composed of rectangular cells within which targets are randomly distributed. Targets are constrained to move only within a cell with a constant velocity corrupted with a white noise measurement. Each asset's capabilities and performance is modeled using a Kalman filter observation matrix (one for each sensor) and noise variance of their measurements. The sensor management scenario is modeled as a market-based system where the consumer agents bid for and purchase various sensor resources from the sensor manager. Consumer agents are assigned the responsibility of searching for and destroying targets. They are required to reduce the error covariance of the target estimates below a mission specified value before destroying them. MASM uses explicit service mappings for this simulation framework.

Implementation details

Target Modeling: Targets are randomly distributed throughout the search area. For this initial version, targets are constrained to move only within a cell with a constant velocity corrupted with a white noise. The target motion is simulated by the equation

$$x_k(t+1) = x_k(t) + w_k,$$

where $x_k(t+1)$ and $x_k(t)$ are positions of the target at time t+1 and t respectively, and w_k is the white noise with known constant covariance Q.

Sensor Modeling: For this simple scenario, sensors can be made to point in four different directions and have two modes of operation: high resolution and low resolution scan. In the low resolution scan mode, sensors can look at a larger area but with lower resolution and vice versa. Sensors are distributed throughout the simulation environment such that all the environment grids fall under the purview of at least one sensor. Each sensor's capabilities and performance are modeled through a Kalman filter observation matrix (one for each sensor) and noise variance of their measurements. The measurement matrix for the i^{th} sensor is given by

$$z_i(k) = x_i(k) + v_k,$$

where $x_i(k)$ is the state vector and v_k is zero mean white noise with known variance R_i.

Consumer Agents: For our initial simulation, the environment is populated with a set of consumer agents, each of whom is assigned an area within which they are responsible to search and destroy targets. Consumers have a smaller area of influence, wherein they can directly pursue and destroy targets. After destroying targets within an influence area, the consumers move to the next unexplored area within their area of responsibility. Consumer agents send bids on various sensor services to the sensor manager and use the assigned services to update the target position information. They are allowed to destroy a target if they successfully bring down the error covariance of the target position measurement below a threshold value, P_{goal}. It is assumed that consumers have a utility U_d for destroying a target.

Sensor Manager: The sensor manager collects bids on sensor services from various consumer agents and determines both the mode of operation and scan direction for each individual sensor. It also maintains the target probability distributions for each cell, which are updated with every sensor measurement. This information is used to provide target state estimates to interested agents.

Information-Theoretic Bidding Formulation: For every cell in their area of influence, consumer agents bid the expected information gain for each possible sensor allocation. The amount of information gained can be measured by the change in entropy prior to and proceeding a sensor measurement. Assuming a normal distribution of target locations, the information gain (i.e., the difference between the a priori and a posteriori entropies) is calculated as

$$I = \log_2(\sigma_s / \sigma_b)$$

where σ_s is the error covariance of the distribution and σ_b is the error covariance before the measurement.

Utility Formulation using Service Chart: For utility estimation, we have not trained a neural network since the sensor network is relatively small. Instead, the service chart calculates the utility of each service possible for a consumer bid online, as follows. A sensor service *S* is defined as a set of sensor measurements on any particular cell. For every consumer bid that the service chart receives, the utility for a particular service is calculated as follows. For example, let a consumer bid a utility of U_d for tracking a target to below the threshold error covariance P_{goal}. For any given sensor service *S*, let *n* be the number of continuous such services required to bring the error covariance of the target estimate below P_{goal}. Hence, the *n* consecutive allocations of service *S* are recovered before the task requested by consumer can be accomplished. In this simulation, we simplify the utility calculation for an individual scan as U_d/n. Thus if a particular target requires three consecutive low resolution scans before the error covariance of its state vector is reduced below the threshold value, then the utility of a consumer agent for a sensor service consisting of a single low resolution scan is $U_d/3$. This utility calculation can easily be made more sophisticated. For example, one can imagine that if you have a heat seeking missile, the first scan imparts the most value (gets you to the right area) and then if you have time, more scans simply add incremental value. The overall detection utility in an unexplored cell for any service is obtained by multiplying U_d by the prior probability of target existence, as calculated by the consumer.

Results

We compared the performance of the information-theoretic sensor manager, utility-based sensor manager, and a randomly scheduling sensor manager. The random sensor manager serves as a baseline and randomly schedules scan directions and scan modes for various sensors. In the low resolution scan, a sensor can scan an area of four cells, whereas in a high resolution scan mode, it can scan only one cell. Table 2 shows the parameters used in the simulation and for the genetic algorithm. We also performed various experiments, varying different parameters including the number of sensors, consumers and girds and obtained similar results.

Simulation Parameters	
No of Grids	64
No of targets	12
No of consumers	8
No of sensors	8
Measurement error covariance for High Resolution scan	$2*10^{-3}$
Measurement error Covariance for low resolution scan	$4*10^{-3}$
Process error covariance	10^{-3}
GA Parameters	
Number of generations	50
Population size	50
Mutation Probability	0.01
CrossOver Probability	0.9
Tour Size	4

Table 2: Simulation and GA parameters

When the threshold for the maximum error covariance of a target allowed before it can be attacked, P_{goal} is set relatively high (i.e., 3 or 4 x 10^{-3}), then fewer scans are needed to acquire enough information to successfully destroy a target. Under these circumstances, both information-theoretic and decision-theoretic approaches destroy targets at the same rate, which is much higher than that demonstrated by the random scheduler. However, when P_{goal} is relatively low (e.g., 1.8 or 2 x 10^{-3}) and more scans/accuracy are necessary, the utility-based approach outperforms the other two techniques, as shown in Figure 5.

Our utility-based sensor manager takes the value of P_{goal} explicitly into consideration in the decision-making process. In contrast, entropy calculations that are used to formulate consumer bids in the information-theoretic sensor manager are independent of the value of P_{goal}. Instead, information-theoretic sensor manager concentrates only on maximization of information gain with every sensor resource allocation. This means that it can oscillate between targets without collecting enough information on one target long enough to successfully destroy it. From this study, we find that using both the information gain *and* the value of that information to the mission (i.e., its utility) to allocate sensor resources is the key to better performance.

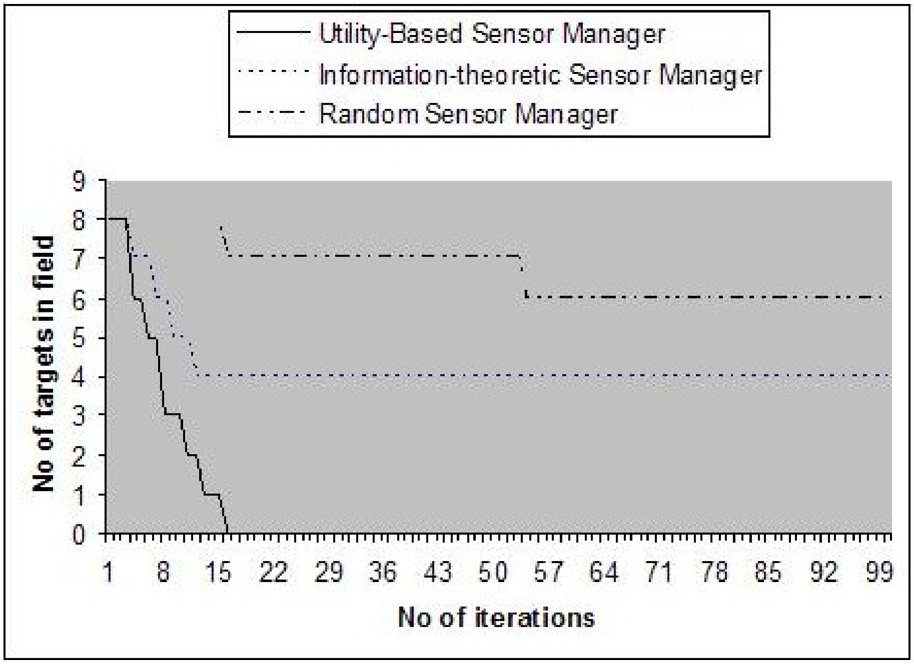


Figure 5: Shows number of iterations before all targets destroyed when P_{goal} is high

Future Work

Market-based approaches provide a valuable framework for designing systems that must consider complex tradeoffs in their decision making. However, they bring with them the challenges of eliciting values for context-dependent situations and the ever-present possibility of participants "gaming" the system for their benefit. Additionally, most current auction research must be extended to trade information as well as tangible resources, and to be truly distributed across multiple sensor management platforms. Below we address our future approaches to these challenges.

Eliciting values for context-dependent situations can be done explicitly (ask an expert) or implicitly (learning). We plan to apply both approaches. While the service chart and bid formulator currently rely heavily on acquiring explicit expert knowledge, we plan to investigate how to learn valuable situation assessment cues from the market bids and price information in the system. The situation assessment that consumers perform typically relates only to their immediate surroundings and pertains to local information only. A global perspective can be obtained by observing the overall market trends in the sensor manager's situation assessment module. For example, a sudden increase in the volume of bids from the consumers in one particular region of the environment could suggest an impending enemy attack in that region. Current prices can convey information about when resources are in high demand and/or scarce. Finally, collaborative filtering approaches can be used to mine information from a combination of goal and bid behaviors.

The potential for "gaming" the system is always a concern in a market-based system, but we believe that it is reduced by two factors. The first is that we operate in a semi-cooperative environment, while all the participants may have individual goals, they are also on the same “team”. We also believe that there is enough disengagement between the actual user decision and the eventual bidding to reduce this behavior. While the service charts and bid formulator modules were originally designed in part to reduce human cognitive load, they also have the potentially positive side-effect of making the actual market process less transparent and harder to game.

Current auction algorithms are generally designed to handle the allocation of tangible e-commerce goods. However, we must adapt these e-commerce algorithms to deal with information products in the sensor-fusion domain. Information products (e.g., observations/reports), and the sensors/processes that generate them, may be shared between agents to effectively complete compatible tasks, where applicable. For example, if two consumers are engaged in the same sub-goal, then a single commodity can be communicated to both agents and will satisfy both of them. Also, domain-specific product mappings can be used to increase sensor sharing. For example, in the case of receivers, if consumer A is interested in scanning an area at rate r_1 and consumer B is interested in scanning the same area at rate $r_2 > r_1$, then it may be possible that a single information product (i.e., a scan at rate r_2) can be communicated to both agents and will satisfy both of them.

Finally, we are extending our current single platform work to handle a more comprehensive system of distributed sensor systems by incorporating distributed auction design. We plan to evaluate auction designs based on different communication and real-time constraints, and identify what auction mechanisms and protocols respond best under different sensor management domain conditions. For example, in highly communication-bounded scenarios, auction protocols with high communication overhead would not be appropriate.

The rapid evolution of advanced sensors, network centric concepts, and use of web-based service implementation of distributed sensing/processing systems motivates the investigation of ecommerce approaches to sensor management. As smart sensors become imbedded in everyday products, wide-band communications enables world-wide distribution of sensed information, an increasing need exists to determine how to link information users (consumers) with information suppliers (smart sensors). Advances in the technology of sensors, sensor networks, and human-in-the-loop information users provide both a challenge (viz., how to optimally use information resources to satisfy the needs of users) and an opportunity (the ability to use market-based concepts for dynamic allocation and re-allocation of system resources to satisfy the system users. Our research provides a beginning approach to model such systems and to compare the effectiveness of market-based algorithms for resource allocation.

Acknowledgements
This research was funded in part by a grant from the U.S. Army Research Office for Center of Excellence for Battlefield Sensor Fusion (via the Tennessee State University). We would also like to thank Dr. Richard Sherry for providing information about sensor node performance.

References

[1] C. G. Schaefer and K. J. Hintz, "Sensor management in a sensor-rich environment," presented at Proceedings SPIE, Signal Processing, Sensor Fusion, and Target Recognition IX, Orlando, FL, 2000.

[2] P. D. Scott and G. L. Rogova, "Crisis Management in a Data Fusion Synthetic Task Environment," presented at The 7th International Conference on Information Fusion, Stockholm, Sweden, 2004.

[3] A. Nandan, S. Das, G. Pau, M. Gerla, and M. Y. Sanadidi, "Co-operative Downloading in Vehicular Ad-Hoc Wireless Networks," presented at Wireless on demand Network Systems and Services (WONS 2005), 2005.

[4] D. S. Alberts, J. J. Garstka, and F. P. Stein, *Network Centric Warface: Developing and Leveraging Information Superiority*, 2nd Edition (Revised) ed: CCRP publication series, 1999.

[5] D. E. Culler and H. Mulder, "Smart Sensors to Network the World," in *Scientific American*, 2004, pp. 84-91.

[6] R. Govindan, E. Kohler, D. Estrin, F. Bian, K. Chintalapudi, O. Gnawali, S. Rangwala, R. Gummadi, and T. Stathopoulos, "Tenet: An Architecture for Tiered Embedded Networks," University of Southern California CENS Technical Report 56, November 10 2005.

[7] M. P. Wellman, "A market-oriented programming environment and its application to distributed multicommodity flow problems," *Journal of Artificial Intelligence Research*, vol. 1, pp. 1-23, 1993.

[8] J. M. Manyika and H. F. D. Whyte, *Data fusion and Sensor Management: A Decentralized Information Theoretic Approach*. New York: Ellis Horwood, 1994.

[9] A. Steinberg and C. L. Bowman, "Revisions to the JDL Data Fusion Model," in *Handbook of Multisensor Data Fusion*, D. L. Hall and J. Llinas, Eds. Boca Raton, FL: CRC Press, 2001, pp. 2-1:2-18.

[10] T. Mullen, V. Avasarala, and D. L. Hall, "Bringing the Market to Level 4 Fusion: Investigation of Auction Methods for Sensor Resource Allocation," presented at MSS National Symposium on Sensors and Data Fusion, Monterey, CA, 2005.

[11] K. J. Hintz and J. Malachowski, "Dynamic goal instantiation in goal lattices for sensor management," presented at Proceedings of SPIE, 2005.

[12] F. Dormon, V. Leung, D. Nicholson, E. Silva, and M. Williams, "Information-based decision making over a data fusion network," presented at Proceedings of SPIE.

[13] K. Veeramachaneni, L. Osadciw, and P. K. Varshney, "An adaptive multimodal biometric management algorithm," *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 35, pp. 344-356, 2005.

[14] P. D. Ezhilchelvan, S. K. Shrivastava, and M. C. Little, "A model and architecture for conducting hierarchicaly structured auctions," presented at The fourth International IEEE Symposium on Object oriented Real-time Computing (ISORC), Magdeburg, Germany, 2001.

[15] F. Ygge, "Market-Oriented Programming and its Application to Power Load Management," in *Department of Computer Science*. Sweden: Lund University, 1998.

[16] D. C. Parkes and J. Shneidman, "Distributed Implementations of Vickrey-Clarke-Groves Mechanisms," presented at Proc. 3rd Int. Joint Conf. on Autonomous Agents and Multi Agent Systems, 2004.

[17] R. V. Denton, E. Alcaraz, J. Knopow, J. Llinas, and K. Hintz, "Towards Modern Sensor Management Systems," presented at Sixth Joint Service Data Fusion Symposium, Washington, DC, 1993.

[18] R. Muraleedharan and L. Osadciw, "Decision in a Building Access System Using Swarm Intelligence and Posets," presented at Conference on Information Sciences and Systems, Princeton, NJ, 2004.

[19] G. A. McIntyre, "A comprehensive approach to sensor management and scheduling." Fairfax, VA: George Mason University, 1998.

[20] T. Sandholm, S. Subhash, A. Gilpin, and D. Levine, "CABOB: A fast optimal algorithm for winner determination in combinatorial auctions," *Management Science*, vol. 51, pp. 374-390, 2005.

[21] H. Hoos and C. Boutilier, "Solving combinatorial auctions using stochastic local search," presented at American Association of Artificial Intelligence, 2000.