A Design for Distributed Collaborative Adaptive Sensing of the Atmosphere

D. Pepyne, D. Westbrook, B. Philips, E. Lyons, M. Zink, and J. Kurose

Abstract—**Consisting of a tightly meshed network of short range Doppler weather radars, Distributed Collaborative Adaptive Sensing (DCAS) represents a new paradigm in meteorological sensing. Rather than sensing everything in a sit-and-spin manner, a DCAS system is an end-user driven sensor network that uses targeted sector scans to sense only those volumes that the end-users have indicated are important to their data needs. The advantage is that by limiting what is sensed, higher quality measurements are possible due to the ability to dwell longer, scan more elevations, obtain better multi-Doppler coordination, and sample at higher rates. This paper describes the Meteorological Command & Control (MC&C) component in the first prototype DCAS system recently fielded by the National Science Foundation sponsored Engineering Research Center for Collaborative Adaptive Sensing (CASA-ERC).**

I. Introduction

The Engineering Research Center for Collaborative Adaptive Sensing of the Atmosphere (CASA-ERC) seeks to revolutionize our ability to observe, understand, and predict hazardous weather by creating distributed collaborative adaptive sensing (DCAS) networks that sample the atmosphere where and when end-user needs are greatest [20]. Configuration-wise, DCAS refers to the use of large numbers of short-range Doppler weather radars appropriately spaced to achieve extensive overlapping coverage. Short range reduces resolution degradation due to beam spreading and by avoiding blockage due to the curvature of the Earth allows the radars a clear view of the bottom 1km of the troposphere (80% of which is blocked from the view of today's NEXRAD weather observing network) where tornados and other weather hazards form. Operations-wise, extensive overlapping coverage allows the radars to cooperatively and collaboratively share responsibility for detecting and tracking the weather passing through the network. Specifically, instead of sensing everything in a sit-and-spin manner as is done in most of today's weather sensing networks, a DCAS network uses *targeted sector scanning* to sense only those volumes of the atmosphere that are most important to the information needs of its end-users. That is, unlike most sensor networks that "push" the same data to all end-users, DCAS networks feature data "pull" where end-user preferences and information needs drive the allocation of the sensing resources [16, 28]. By limiting what is sensed it is possible to obtain higher sample rates, cover more elevations, and get improved sensitivity [8].

A series of testbeds, known as Integrative Projects (IPs), are being fielded by the CASA-ERC to demonstrate and test the DCAS concept. The first, the four-radar IP1 network, became operational in August 2006. Strategically located in southwestern Oklahoma's "tornado alley" in a region that receives a yearly average of 4 tornado and 53 thunderstorm warnings, a major goal of IP1 is to understand the impact DCAS can have on the ability of the IP1 end-users to anticipate, detect, track, and respond to these severe weather hazards. End-users of the IP1 data include the National Weather Service (NWS) forecast office in Norman Oklahoma whose role is to issue severe weather watches and warnings; a group of regional Emergency Managers (EMs) in and downstream of the testbed whose role is to alert the public and coordinate first responders; and CASA researchers working on a variety of projects including improved forecast models, detection algorithms, radar technologies, and sensor allocation algorithms [23]. Some of these users will be interpreting the radar data through visual displays. Other "users" are numerical algorithms that will be performing signal processing operations on the data. Thus, at any given instant, each group of end-users will have different, potentially conflicting, requirements for what volumes of the atmosphere they need to have scanned.

The purpose of this paper is to describe the Meteorological Command and Control (MC&C) component in the IP1 Oklahoma testbed. As the main control loop of the IP1 network, the MC&C is responsible for adapting and configuring the radar beam steering commands to sense where and when the dynamically changing information needs of the end-users are greatest [16, 23, 28]. The control challenge of DCAS targeted sector scanning is that time bounds on how fast a volume can be scanned – in general the slower we sweep a radar beam the better the data quality [9] – introduces both intra-user and inter-user resource conflicts. Such resource conflicts arise for example when one user requires scanning a large volume while another requires two or more radars to collaborate to pinpoint narrowly on a wind event passing through a region of the network where multiple of its radars overlap to obtain triangulated velocity vector measurements. Given the

This work was supported primarily by the Engineering Research Centers Program of the National Science Foundation under NSF award number 0313747. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.

All authors are with the CASA Engineering Research Center at the University of Massachusetts at Amherst, Amherst, MA 01003 USA. J. Kurose is also a Professor in the Dept. of Computer Science at the Univ. of Massachusetts, Amherst.

Corresponding author, D.L. Pepyne (email: pepyne@ecs.umass.edu).

coverage pattern of the IP1 network – see Fig. 2, also [3, 21] – this type of resource conflict would be particularly acute in the event of a confirmed tornado passing through the center of the network. For EMs, it is critical to obtain specific information about the tornado's location, the path it has taken so far, and its anticipated future track. In this unprecedented case of a tornado passing through a four radar multi-Doppler coverage region, CASA's researchers would demand a multi-Doppler scan with as many radars as possible at the maximum sample rate the network can provide. Depending on factors such as attenuation, this may require focusing all four of the IP1 radars on the volume containing the tornado [2]. In contrast, the NWS – once they have issued a warning on the tornado – will need to scan the structure of the potentially very large volume of the tornado's parent storm as well as the volumes of any other surrounding storms to assess the chance additional tornados might be forming elsewhere in the network. In this case the conflict is clear, depending on how the radars are allocated, the data quality provided to one or even all groups of end-users can suffer. For additional details regarding the IP1 end-users and their needs see Section III and [23].

To handle resource conflicts such as the one above, the IP1 MC&C employs an approach that combines expert rules to represent end-user scanning requirements with multi-attribute utility/score functions to turn the rules into a mathematical form that we can optimize over to make the tradeoffs necessary to preferentially allocate the sensing resources towards those scanning tasks that the end-users have collectively agreed are the most important to satisfy. Rules have long been a standard way to represent expert knowledge (in our case here, what to scan and how to scan it) [10]. Scoring is a common way to assess tradeoffs and rank options in combinatorial optimization problems (in our case here the ranking of the alternative scanning options) [24, 27]. And as a systematic method for decision problems involving multiple-users with multiple competing preferences and objectives multi-attribute utility theory (MAUT) [15] is becoming increasingly popular for allocating resources in sensor networks, cf. [1, 7, 17].

The remainder of this paper is organized as follows. Section 2 describes the closed-loop IP1 Command & Control architecture. Section 3 presents rules that represent the scanning preferences and needs of the IP1 end-users. Section 4 details the algorithms in the MC&C that map the end-user rules into radar sector scans. Performance results collected during a storm event in the IP1 testbed are presented in Section 5. The paper concludes with a summary and discussion of on-going work in Section 6.

II. IP1 Command & Control Architecture

Fig. 1 shows a schematic of the IP1 Meteorological Command & Control (MC&C) architecture. As the main control loop, the MC&C is responsible for mapping end-user needs and past radar observations into a configuration of radar sector scans that optimally trades off resource conflicts to maximize end-user satisfaction [5, 16, 28]. The IP1 MC&C can be described as discrete-time control system operating on a 30 second "heartbeat." Every 30 seconds the system goes through a process of data ingest, meteorological feature detection, task generation, and optimization to choose the sectors the radars will scan during the next heartbeat.

A. Data Ingest and Feature Detection

Starting with the radars, the IP1 testbed consists of four mechanically steered, parabolic dish, X-band Doppler radars spaced nearly equidistantly 25 km apart in Cyril, Chickasha, Rush Springs, and Lawton Oklahoma. With an approximate 30 km range each, this spacing was chosen to maximize the amount of overlapping coverage [3, 21].

Each of the IP1 radars generates about 100 Mbit of raw time-series data each second. A Fourier transform is performed at each radar node to extract the spectral moments (moment data) from the time series data; reducing the data output from each radar to about 4 Mbit of data each second. For Doppler weather radars, the 0^{th} moment (the reflectivity) is related to the water content in the atmosphere, the 1^{st} moment gives the wind speed toward or away from the radar, and the 2^{nd} moment gives the velocity dispersion which measures the wind sheer or turbulence [9].

The moment data are transmitted from the radar nodes to the IP1 Systems Operation and Control Center (SOCC) at the Univ. of Oklahoma in Norman, where it is processed by meteorological feature detection algorithms to identify such meteorological features as storms (regions of high reflectivity), strong winds (regions of high velocity), and rotations (correlated velocity vectors) [12]. The moment data along with the detected weather features are transformed from the polar coordinates of the radars to a common lat-long coordinate system, merged with other GIS data (e.g., manually entered locations of reported tornado sightings), and posted on a blackboard architecture known as the *feature repository*, where it is made available to the end-users The use of a blackboard architecture for the feature repository was chosen to make the system tolerant to data ingest problems (e.g., radar failures and communication losses) and to make the system scalable (by distributing the feature repository) as additional radars are added to the network [6, 14].

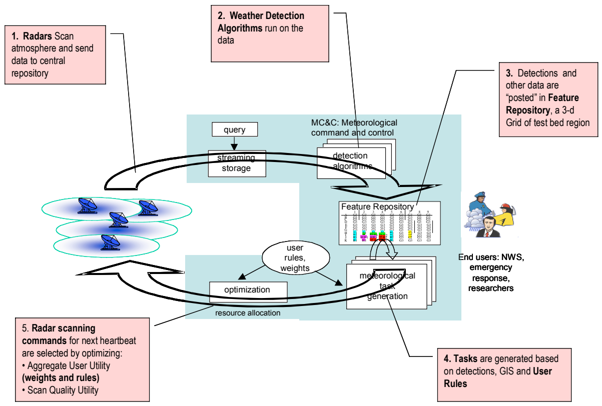

Fig. 1. Schematic showing the flow of data and control within IP1.

B. Task Generation

The features in the feature repository are clustered (via K-means) based on similarity metric that combines lat-long location and feature type (storm, wind, rotation). To make the cluster centers stable from one heartbeat to the next, we use the cluster centers from the previous heartbeat to initialize the K-means algorithm. The feature clusters output from the clustering algorithm are defined by a type and a polygon in 2-d lat-long coordinates formed by taking the convex hull of the features that make up the cluster.

Scanning tasks are generated by matching the feature clusters against a set of "if-then" rules that define the end-user requirements for what to scan, when to scan it, and how to scan it (see Section III). For each feature cluster matching the "if" part of a rule we generate a unique *scanning task*. Thus, we note that each end-user rule can generate several scanning tasks each located at a different place in the network. Moreover, different end-user rules can require the same feature cluster be scanned in different ways, e.g., using different scan rates. A scanning task is a tuple consisting of the 2-d area in lat-long coordinates where the task is located and the rule that tells when and how to scan the task. In addition to the "tracking" tasks generated by feature detections, each user may also have one or more periodic time based rules that generate, for example, periodic 360 degree surveillance tasks to look for emerging weather.

C. Optimization

As mentioned, an innovation of a DCAS network is its use of targeted sector scans. In the IP1 network, the radars are mounted on mechanically steered pedestals that rotate at 28 degrees/second in azimuth and can be pointed to 7 discrete elevation angles. With every 30 second heartbeat, each radar is commanded to scan a single sector, where a sector is a wedge in azimuth of a certain angular width and compass orientation. Each radar will start with a horizontal sweep of its sector at its lowest elevation tilt and will sweep back and forth, working up through as many discrete elevation tilts as it can in the 30 second sampling interval. Hence, while a radar can scan its complete 360 degree area at the lowest two elevations in a 30 second heartbeat, it is limited to sectors of no more than 120 degrees if required to obtain a full 7 elevation scan of a volume. An IP1 radar can thus only scan 1/3 of its surrounding volume in single heartbeat.

To determine the optimal configuration of sectors for the four radars to scan, the MC&C uses a utilities/score based approach. Two utilities are assigned to each scanning task – a utility score U telling *how important* the task is to the end-users, and a scan utility function Q-function that relates *how well* a particular configuration of sectors meets the scanning requirements of the rules associated with the tasks. Optimization of an overall utility function, called the MC&C equation, combines these two factors of how important with how well to determine the configuration of sectors that can be expected to satisfy the maximum number of the highest value tasks (see Section IV).

Fig. 2. IP1 operators interface showing weather features (yellow polygons), scanning tasks (brown polygons), and sectors (shown in green, with the rings along the sector edge denoting the number of elevation angles in the scan).

Fig. 2 shows a screen shot of the IP1 operator's interface; detected features are in yellow, scanning tasks are in brown, the sectors scanned are in green (the rings along the sector edge denote the number of elevations scanned).

III. Representing End-User Data Needs

Table I shows the current set of "if-then" rules representing the data needs of the IP1 end-users – National Weather Service (NWS) forecasters, Emergency Managers (EMs), and CASA Researchers. In the table, the "trigger" column is the "if" part of the rule, indicating the event that invokes the rule. *Time* based rules are invoked with the expiration of a timer; *storm*, *wind*, and *rotation* rules are invoked by the detection of the indicated type of feature cluster in the feature repository (recall Section II). The remaining columns in the table are the "then" part of the rule identifying the scanning constraints that must be met in order to satisfy the rule. The "sector" column identifies a *coverage constraint* that specifies how much of a task must be covered in order to satisfy a task. This is a hard constraint, a task is not satisfied if it is not completely covered in azimuth. This constraint (combined with the minimum 60 degree sector width) ensures that the complete context of storms, including their boundaries, are included in the scan data. This is particularly important for visual interpretation of radar data [23]. The "elevations" column identifies an *elevation constraint*. This is a soft constraint; task satisfaction increasing with the number of elevations scanned relative to the number required by the rule. Multiple elevations are needed to understand a storm's vertical structure, such as the helical rotations in tornados. The "#radars" column indicates a constraint on the number of radars that need to be assigned to the task. Generally, reflectivity can be adequately measured with a single radar. However, because a single radar can only measure velocity in a single direction, accurate velocity vector estimation requires triangulating multi-Doppler views from two or more linearly independent directions. The *#radars constraint* is a soft constraint; scan satisfaction for tasks requiring more than one radar increasing as more radars are assigned to the task. Finally, the "revisit" column indicates a *sample rate constraint*. This is based on the typical dynamics of various weather features – their horizontal movement, evolution in size, and evolution of internal structure.

TABLE I
END-USER RULES FOR IP1.

Rule	Trigger	Sector	Elevations	#Radars	Revisit
NWS – issue watches and warnings.					
N1	Time	360	Lowest 2	1	1 / min
N2	Storm	task	All 7	1	1 / 2.5 min
Researchers – numerical weather prediction, algorithm, model, radar development					
R1	Rotation	task	All 7	2+	1 / 30 sec
R2	Storm	task	All 7	1	1 / min
R3	Time	360	All 7	1	1/ 10 min
Emergency Managers – public notification, spotter/first responder deployment.					
E1	Time	360	Lowest 2	1	1 / min
E2	Storm	task	All 7	1	1 / min
E3	Wind	task	All 7	2+	1/ 2.5 min
System Operator – data assimilation, resource allocation experiments					
O1	Time	360	Lowest 2	1	1 / 5 min
O2	GUI input	Task	User defined, Lowest 2 or all 7	User defined, 1 or 2+	Scan once

The rules in Table I were elicited from the IP1 end-users through a process involving a review of historical best practices, in-depth interviews with subject matter experts, and table-top demonstrations with simulated data [23]. NWS rule N2 for example indicates that storms should be scanned at all elevations with a sector that covers the entire storm at a sample rate of once every 2.5 minutes. This rule addresses the NWS forecaster need to analyze vertical storm structures as they are determining whether to issue warnings. The Researcher Rule R3 to execute a complete scan of the entire volume covered by the IP1 network reflects a need for data to initialize numerical weather prediction models. Recalling that a radar can only scan 1/3 of its surrounding volume in a single 30 second heartbeat, Researcher Rule R3 will require several heartbeats to complete. To accommodate this rule, we partition the volume around each radar into three 7 elevation, 120 degree sector scanning tasks, each with a once per 10 minute sample rate requirement. Regarding the system operator rules, O1 is for data assimilation to maintain a history of the weather passing through the network (this rule becomes necessary only when the user 360 rules (N1 and E1) are "turned off", e.g., by zeroing the user priority weight as described in Section IV); O2 provides a two-way interface through which an operator can manually insert scanning tasks, define their sizes and specify their scanning requirements (e.g., to scan an area where a tornado has been spotted).

IV. THE MC&C EQUATIONS

Looking at the rules in the previous section we can reduce a scanning task into one of four basic types – tracking tasks requiring single radar, multiple elevation sector scans; tracking tasks requiring multiple radar, multiple elevation sector scans for velocity vector triangulation; 360 surveillance tasks of the lowest two elevations; and data assimilation tasks to scan of the entire 360 degree 7 elevation volume around each radar. Given that a radar can only scan 1/3 of its surrounding volume in any single system heartbeat, resource conflicts will be determined by the number and location of these different types of tasks. Specifically, while a sector can be widened to cover more tasks, this can reduce scan quality by reducing the number of elevations that can be covered (conflict between the coverage and elevations constraints). Alternatively, while the individual radars can coordinate to point in different directions to maximize the number of single radar tasks they can collectively satisfy, where they point is constrained to certain compass directions when collaboration is required on multiple radar tasks (conflict between single and multiple radar tasks, recall the radar coverage map in Fig. 2). This section describes the five step process of – time since last scanned determination, task utility score assignment, task scan quality function assignment, utility optimization, and task satisfaction updating – that maps the end-user scanning requirements into the radar sector assignments.

A. *Time Since Last Scanned Determination*

A rule's "revisit" column gives a task's desired sample rate. To determine if a task needs to be scanned during the next heartbeat, we first need to determine the time since it was last successfully scanned. For the periodic "360" tasks (those associated with rules N1, R3, E1, and O1) keeping track of the time since the task was last satisfied is trivial –

just keep track of the last time each radar did a 360 degree scan of the lowest two elevations. For tasks generated by the feature detection algorithms (those associated with rules N2, R1, R2, E2, and E3), the process is not so simple. For these tasks, we maintain a rolling history of successfully scanned tasks, which includes the 2-d polygon scanned, the rule that was satisfied, and the heartbeat k_s when the task was last successfully scanned. To account for storm movement and growth, we use a lightweight "storm tracker" that works by simply moving the corners of the polygons outward from their geometric centers at a rate of 55km/hr. A newly generated task is considered an old one if it falls inside the expanded polygon of a previously scanned task associated with the same rule, and we set its time last scanned to k_s. Otherwise the task is a new one and we set k_s to minus infinity (to indicate that this task has not yet had a successful scan). While this tracker has so far proven adequate for our purposes, it can be upgraded if needed with, for example, [19].

B. Task Utility Score Assignment

With g the index of the user group whose rule generated the scanning task and k the current heartbeat, we next assign to each task t a task utility score $U_g(t,k\text{-}k_s)$ that depends on the number of heartbeats since the task was last scanned $k\text{-}k_s$ relative to the "revisit" interval r_t (in heartbeats between scans) required by the task's rule. Specifically, we set the task utility score to one of three values,

1. $U_g(t,k\text{-}k_s) = 0.3$ if $(k\text{-}k_s) < r_t$ (i.e., the task does not need to be scanned at the next heartbeat);
2. $U_g(t,k\text{-}k_s) = 0.8$ if $(k\text{-}k_s) = r_t$ (i.e., the task needs to be scanned at the next heartbeat);
3. $U_g(t,k\text{-}k_s) = 1.0$ if $(k\text{-}k_s) > r_t$ (i.e., the task is overdue for scanning).

A task utility of 0.3 ensures the system does something even when there are no tasks immediately in need of scanning. The idea behind raising the utility of overdue tasks to 1.0 is the following. Because of the relatively slow dynamics of weather, only tornados need to be sampled at the maximum twice per minute sample rate of the IP1 system. All other tasks can be sampled at much lower rates. Thus, by raising the utility of unsatisfied tasks to 1.0 we can force the system to try to deconflict competing tasks by scheduling them to be scanned on interleaving heartbeats. This idea is inspired by PD (proportional plus derivative) control – use feedback to track a task's required sample rate. An extension being considered would introduce an "integral" term for a PID-like scheme that decreases the U-value for tasks that are being oversampled, and increases it for those that are being undersampled. The main advantage of this simple feedback approach is that it addresses what is otherwise a very hard multistage optimization problem (see for example [11, 26] for a treatment of acquisition and tracking in sensor networks). See [18] for a stochastic dynamic programming treatment of the CASA DCAS problem.

To get the "community" utility $U(t)$ for each task we weight it by, w_g, the "priority" of the user group whose rule generated the task,

$$U(t) = w_g U_g(t, k - k_s) \tag{1}$$

Taking values between 0 and 1, w_g determines the relative "effort" the system will make to satisfy the tasks generated by user group g. Setting $w_g = 0$ gives user group g zero priority in the system by effectively turning off the group's rules. Larger values of w_g give the user's rules increasing influence on the scans the system selects. Although priority weights are a common way of combining user needs in multi-user systems, getting users to agree on what their individual priority weights should be can be a contentious process, since no one wants to be considered a low priority user. We view (1) as a *mechanism* for setting user priorities. An on-going research project to relate the value of each user group to reducing the public socioeconomic impact of hazardous weather will ultimately be used to develop a *procedure* for setting priorities [23].

C. Task Scan Quality Function Assignment

To measure the degree to which different sector scan configurations satisfy the "coverage," "elevations," and "#radars" constraints of a task's rule, we assign a scan quality function. The construction of this function is a two step process: first the scan quality is determined for each radar acting individually; then the aggregated scan quality is determined by combining the individual scan quality values.

Individual scan quality – The individual scan quality $q(t,r,s_r)$ gives the degree to which the sector s_r scanned by radar r satisfies coverage and elevations constraints of task t's rule. Specifically, let us define: $w(s_r)$ = the azimuthal width (in degrees) of radar r's sector; $a(r,t)$ = the minimal azimuthal angle that would allow radar r to just cover task t's 2- area; $h(r,t)$ = the distance from the radar to the geometric center of the task; and $h_{\max}(r)$ = the range of radar r. Then let us define,

$$q(t,r,s_r) = F_c(c(t,r,s_r))[\alpha F_e(e(w(s_r))/e_r(t) + (1-\alpha)F_d(d(r,t))] \tag{2}$$

where, in terms of the above definitions, $c(t,r,s_r) = w(s_r) / a(r,t)$ is the coverage of task t by radar r with sector s_r; $e(w(s_r)) = 840 / w(s_r)$ is the number of elevations a radar scanning a sector $w(s_r)$ degrees in azimuth at an angular rotation rate of 28 degrees / second can do in a 30 second heartbeat; $e_r(t)$ = the number of elevations required by task t's rule; $d(r,t) = h(r,t) / h_{\max}(r)$ is the normalized distance from radar r to the geometric center of task t; $\alpha \in [0,1]$ is a tunable parameter (set to 0.9 in the current implementation); and F_c, F_e, and F_d are the step functions defined in Fig. 3a-c respectively.

The rationale for Equation (2) is as follows. The first term $F_c(c(t,r,s_r))$ accounts for how well the task is covered in azimuth. Noting that this term multiplies the other terms in equation (2) we see that if the task is not entirely covered in azimuth, the scan quality is zero as per the hard *coverage constraint*. The second term $F_w(w(s_r))$ reflects the soft *elevation constraint* by penalizing scans that don't get all of the elevations requested by the task's rule. The third term $F_d(d(r,t))$ is a soft *range constraint* included to decide which radar(s) to use when the task is in the coverage area of more than one radar. According to $F_d(d(r,t))$, radars closer to a task tend to result in better scan quality due to considerations such as intervening attenuation and resolution degradation caused by increased angular beam spreading with distance (see also [2, 25]).

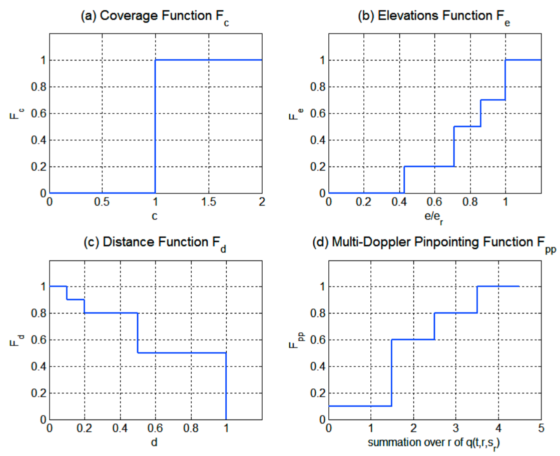

Fig. 3. Definitions of the step functions used in equations (2) and (4).

Combined Scan Quality – The combined scan quality gives the degree to which a scan configuration $C = \{s_r : r=1,\ldots,4\}$, where s_r is the sector scanned by radar r, satisfies the "#radars" constraint of a task's rule. For tasks that require only one radar the combined scan quality is obtained by taking the maximum of the individual scan qualities,

$$Q(t,C \mid \#radars = 1) = \max_{r=1,2,3,4} \left[q(t,r,s_r) \right] \qquad (3)$$

For tasks that require multiple radars we combine the individual scan qualities according to,

$$Q(t,C \mid \#radars = 2+) = F_{pp}\left(\sum_{r=1,2,3,4} q(t,r,s_r) \right) \qquad (4)$$

where the function $F_{pp}(\ )$ is as defined in Fig. 3d. Noting that $q(t,r,s_r) \in [0, 1]$ for each radar r, the interpretation of equation (4) is to give increasing utility for each additional radar that scans the task – the more radars scanning the task, the better the ability to resolve velocity vectors.

D. Utility Optimization

Given $U(t)$ and $Q(t,C)$, the scan configuration C to be used during the next heartbeat is the one that maximizes the objective function,

$$J(C) = \sum_t U(t) I\left[Q(t,C) \geq 0.6 Q_{\max}(t) \right] \qquad (5)$$

where $I[\ ]$ is the indicator function (=1 if its argument is true; 0 otherwise), and $Q_{\max}(t)$ is the maximum scan quality the system could achieve if task t were the only task in the system. $Q_{\max}(t)$ is easily computed from the task location, its 2-d area, and its associated rule. We call equation (5) the *MC&C equation*. The first term $U(t)$ reflects the collective end-user preference for scanning the volume represented by scanning task t at heartbeat k, and letting C^* be the argument that maximizes (5), we say that a task t is satisfied if $Q(t,C^*) \geq 0.6 Q_{\max}(t)$, where 0.6 is an adjustable parameter. Optimizing (5) can thus be interpreted as the preferential allocation of the sensor resources to satisfy those scanning tasks that the end-users have collectively agreed are the most important to satisfy.

E. Time Last Scanned Updating.

After each scan we update the task history. For tasks satisfied under C^* (i.e., tasks for which $Q(t,C^*) / Q_{\max}(t) \geq 0.6$), we set the time last scanned $k_s = k$. For tasks not satisfied under C^* (i.e., tasks for which $Q(t,C^*) / Q_{\max}(t) < 0.6$), we leave k_s unchanged.

V. Performance Analysis

Preliminary experiments were conducted to assess how well the IP1 MC&C design is able to satisfy end-user needs by evaluating how well the system is able to satisfy the rules that define those needs. Data for the experiments was obtained from actual scans of a severe storm that passed through the IP1 testbed between 2:30AM and 5:00AM on 16 August 2006. See [4] for a system level discussion of network operations during the August storm event. This storm was of sufficient severity for the NWS to issue one thunderstorm warning and several severe wind reports. During the experiments velocity data was not available, so feature detection was limited to using only reflectivity. The experiments reported here thus only evaluate the MC&C's ability to satisfy the reflectivity-based rules (N1, N2, R2, R3, E1, E2, and O1 in Table I). The IP1 network is continuously being updated, and the final paper will provide an analysis with the complete set of end-user rules operating.

A. DCAS Sector Scanning Algorithm Performance

Over the 2.5 hours of the experiment there were a total of 2943 tasks submitted to the optimization for scanning, for an average of 10.3 tasks per heartbeat. Of these a total of 1221 tasks, or an average of 4.3 per heartbeat, could not be satisfied during a typical heartbeat due to resource conflicts. As expected, the tasks that the system had difficulty satisfying were those requiring 7 elevation volume scans of spatially isolated sectors, i.e., tasks associated with rules N2, R2, R3, E2, and O1. Specifically, of the average 8.9 such tasks generated per heartbeat, an average of 4.3 (48%) were not satisfied. All of the time based 360 tasks generated by rules N1, E1, and O1 were satisfied at the same heartbeat they were submitted. Thus, over the period of the experiment, there was a 100% chance a 360 task submitted to the system would be satisfied, but only a 52% chance a full 7 elevation sector scanning task would be satisfied at the heartbeat it was submitted.

On the other hand, recall that if a task due to be scanned at a particular heartbeat is not satisfied we increase its utility and continue to resubmit it for satisfaction until such time as it is either satisfied or moves out of the network. A consequence of this strategy, however, is that unsatisfied tasks could begin to accumulate and overwhelm the system. The plot in Fig. 4 shows that this is not happening, meaning that although the system is not able to satisfy every task immediately when it is due to be scanned, the system does eventually satisfy all tasks submitted to the system. In fact, because we record the total delay between the time a task is submitted and the time it is scanned we can estimate the sample rate performance of the system. For tasks associated the with 7 elevation sector scans (N2, R2, E2, and O1), the average sample rate was 55.26 seconds between scans (thus satisfying the once per minute required sample rate of R2 and E2, and more than satisfying the once per 2.5 minute sample rate requirement of N2 and the once per 5 minute requirement of O1). For tasks associated with numerical weather prediction rule R3, the average sample rate was 3.55 minutes between scans, thus beating the once per 10 minute sample rate requirement. The reason rules such as R3 are oversampled is that time-triggered rules are always active, i.e., even when they are not due for scanning they are given a small but non-zero utility score (recall Section IVb). Hence, when there are no high utility tasks due for scanning at a given heartbeat, the system will not sit idle, but will generate scans of these low utility tasks.

B. *Sit-and-Spin Algorithm Performance*

To show the advantages of the DCAS approach of targeted sector scanning we compared its performance to the so-called sit-and-spin scanning algorithm. Sit-and-spin scanning can be viewed as the no control case – the sit-and-spin strategy simply repeating 360° sweeps of the lowest 2 elevations with every heartbeat. The results were obtained by replaying the tasks generated during the 16 August 2006 storm event through the MC&C while we operated it in sit-and-spin mode. Except for the fact that we did not use the output of the optimization to generate the beam steering commands, sit-and-spin went through all the same steps of task generation, task utility assignment, and resubmission of unsatisfied tasks as used by our sector scanning algorithm.

Over the 2.5 hours of the storm, 8287 tasks – or an average of 28.1 tasks per heartbeat – were submitted to the sit-and-spin algorithm for satisfaction. Of these, only 7% were satisfied at any given heartbeat. As expected, only the 360 tasks (rules N1, E1, and O1) were satisfied. Because sit-and-spin never tilts beyond the second elevation, no task requiring a scan of all 7 elevations could be even partially satisfied (due to the *elevation constraint* in Section IV). The resubmission of these unsatisfied tasks from one heartbeat to the next explains why sit-and-spin had so many more tasks than the sector scanning algorithm.

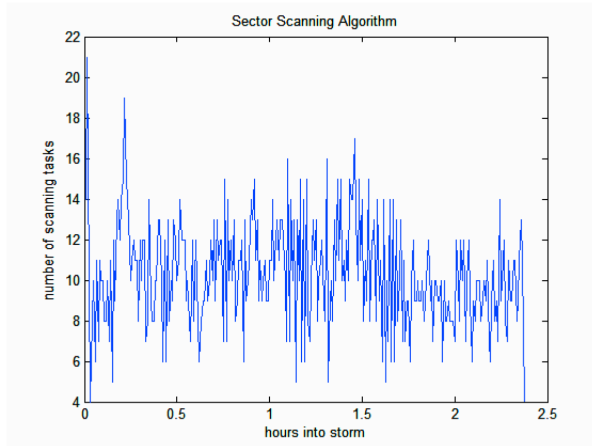

Fig. 4. Number of tasks in the system at each heartbeat.

VI. Conclusions

This paper described an architecture and algorithms for distributed collaborative adaptive sensing (DCAS) of the atmosphere. The architecture is a closed-loop feedback design that combines the real-time context (feature detections and other GIS inputs) with expert defined knowledge (rules) to generate an optimization problem on-the-fly. Solving the optimization problem gives a preferential allocation of the sensor resources to get the best scans of the tasks that the end-users have indicated are the most important to scan. This is one of the primary advantages of DCAS sensing – rather than sensing everything in a sit-and-spin manner, sense only those phenomena that the end-users have indicated are important to their data needs. By limiting what is sensed, higher quality measurements are possible due (e.g., due to the ability to dwell longer, scan more elevations, obtain better multi-Doppler coordination, and sample at higher rates).

The MC&C design described in this paper deals primarily with the problem of deciding where to point the radars in a DCAS network. Exciting ongoing work within the CASA MC&C group includes distributing the optimization which is currently centralized; adding sensing criteria such as signal-to-noise (SNR) feedback to pick the least attenuated radar(s) for a scanning task; incorporation of other sources of GIS information such as NEXRAD detections; adding nowcasting for short term predictive capabilities; addition of dual-PRF capabilities to the radars for improved detection performance; and the ability to adapt dwell time (via varying azimuthal scan rate) in response to the observed weather. Also being explored is the design of revolutionary new network-centric scanning strategies that exploit capabilities that can only be realized by coordinated multi-Doppler scanning such as network based attenuation correction and velocity estimation (cf. [13]); and MC&C designs for future CASA IPs, which will use advanced phased array radars under development by CASA with their "zero intertia" instantaneous beam pointing and their ability to obtain high quality even at high scan speeds.

With IP1 operational, our end-users are now evaluating the data quality being obtained under the end-user scanning rules. As they become familiar with the new paradigm of targeted sector scanning this will surely suggest new scanning strategies and new rules to execute them. Under a new supplement to the CASA grant, we have also started research to incorporate the socioeconomic value of CASA data into our end user policy and resource allocation algorithms. This is involving the development of an integrated decision model of the end-to-end IP1 system to quantitatively link "upstream" technical capabilities, such as targeted sector scans of the bottom 1km of the troposphere, to their impacts on "downstream" responses such as NWS warning decisions, EM risk communication, public response, and the resulting incremental socioeconomic impacts. This end-to-end model will allow us to identify those DCAS capabilities and end-users that provide greatest socioeconomic value.

REFERENCES

[1] F. Bain, D. Kempe, and R. Govindan, "Utility-Based Sensor Selection," IPSN'06, 2006.

[2] K.A. Brewster, E.C. Fay, and F. Junyent, "How will X-band Attenuation Affect Tornado Detection in the CASA IP1 Radar Network," *AMS 32nd Conf. on Radar Meteorology*, October 2005.

[3] K.L. Brewster, L. White, B. Johnson, and J. Brotzge, "Selecting the Sites for CASA NetRad, a Collaborative Radar Network," 9th *Symposium on Integrated Observing and Assimilation Systems for the Atmosphere, Oceans, and Land Surface* (IOAS-AOLS), San Diego, CA 2005.

[4] J. Brotzge, K. Brewster, V. Chandrasekar, B. Philips, S. Hill, K. Hondl, B. Johnson, E. Lyons, D. McLaughlin, and D. Westbrook, "CASA IP1: Network Operations and Initial Data," 23rd *Conference on IIPS*, 2007.

[5] J. Brotzge, D. Westbrook, and M. Zink, "The Meteorology Command and Control Structure of a Dynamic, Collaborative, Automated Radar Network," 21st *Int. Conf. on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology*, 2005.

[6] D.D. Corkill, "Blackboard Systems," *AI Expert*, Vol. 6, No. 9, pp. 40-47, September 1991.

[7] F. Drews, L. Welsh, D. Juedes, D. Fleeman, A. Bruening, K. Ecker, and M. Hoefer, "Utility-Function based Resource Allocation for Adaptable Applications in Dynamic, Distributed Real-Time Systems," *Proc. of the 18th Int. Symposium on Parallel and Distributed Processing*, 2004.

[8] B. Donovan and D. McLaughlin, "Improved Radar Sensitivity through Limited Sector Scanning: The DCAS Approach," Proc. of AMS Radar Meteorology, 2005.

[9] R.J. Doviak and D.S. Zirnic, *Doppler Radar and Weather Observations*, Academic Press, 1993.

[10] J. Durkin, *Expert Systems: Design and Development*, Prentice Hall, 1994.

[11] Y. He and E.K.P. Chong, "Sensor Scheduling for Target Tracking: A Monte Carlo Sampling Approach," *Digital Signal Processing*, 2006.

[12] K. Hondl, V. Lakshmanan, T.M. Smith, and G.J. Stumpf, "Warning Decision Support System – Integrated Information (WDSS-II) Progress and Plans, 23rd (IIPS) *for Meteorology, Oceanography, and Hydrology*, January, 2007.

[13] E. Insanic and P. Siqueira, "Vector Velocity and Vector Velocity Error Estimates in a Networked Radar System," *in preparation*, 2007.

[14] V. Jaganathan, R. Dodhiawala, and L. Baum, *Blackboard Architectures and Applications*, Academic Press, 1989.

[15] R. Keeney and H. Raffia, *Decisions with Multiple Objectives,* Cambridge University Press, Cambridge, 1976.

[16] J. Kurose, E. Lyons, D McLaughlin, D. Pepyne, B. Philips, D. Westbrook, and M. Zink. "An End-User-Responsive Sensor Network Architecture for Hazardous Weather Detection, Prediction and Response", AINTEC, 2006.

[17] G. Mainland, D.C. Parkes, and M. Welsh, "Decentralized, Adaptive Resource Allocation for Sensor Networks," online.

[18] V. Manfredi, "A Comparison of Myopic and non-Myopic Scanning Strategies in A DCAS Meteorological Sensing Network," in preparation, CS Dept., UMass, Amherst.

[19] V. Manfredi,, S. Mahadevan, and J. Kurose, "Switching Kalman Filters for Prediction and Tracking in an Adaptive Meteorological Sensing Network," Proc. of IEEE SECON, 2005.

[20] D.J. McLaughlin, V. Chandrasekar, K. Droegemeier, S. Frasier, J. Kurose, F. Junyent, B. Philips, S. Cruz-Pol, and J. Colom, "Distributed Collaborative Adaptive Sensing (DCAS) for Improved Detection, Understanding, and Prediction of Atmospheric Hazards," preprints of the AMS 9th Symposium on Integrated Observation Assimilation Systems – Atmosphere, Oceans, Land Surface, 2005.

[21] D.J. McLaughlin, E.A. Knapp, Y. Wang, and V. Chandrasekar, "Distributed Weather Radar Using X-Band Active Arrays," *conference reference*, 2007.

[22] National Research Council, *Weather Radar Technology: Beyond NEXRAD*, National Academy Press, 2002.

[23] B. Philips, D. Pepyne, D. Westbrook, E. Bass, J. Brotzge, W. Diaz, K. Kloesel, J. Kurose, D. McLaughlin, H. Rodriguez, and M. Zink, "Integrating End-User Needs into System Design and Operation: The Center for Collaborative Adaptive Sensing of the Atmosphere (CASA)," AMS, 2007.

[24] D.E.R. Pritchard, *Dynamic Route Planning-Replanning and Retasking of Unmanned Aerial Reconnaissance Vehicles*, AFIT Thesis, 2000.

[25] J. Proud, K.K. Droegemeier, V.T. Wood, R.A. Brown, and L. White, "Optimal Sampling Strategies for Hazardous Weather Detection using Networks of Dynamically Adaptive Doppler Radars," 2005.

[26] M.K. Schneider and C. Chong, "A Rollout Algorithm to Coordinate Multiple Sensor Resources to Track and Discriminate Targets," SPIE, 2006.

[27] B.A. Stiles, *Application of Sequential Auction Techniques to Nonlinear Targeting Assignment for Space-Delivered Entry Vehicles*, Masters Thesis, Rice University, Huston TX, May 2004.

[28] M. Zink, D. Westbrook, S. Abdallah, B. Horling, V. Lakamraju, E. Lyons, V. Manfredi, J. Kurose, and K. Hondl, "Meteorological Command and Control: An End-to-end Architecture for a Hazardous Weather Detection Sensor Network," Workshop on End-to-End, Sense-and-Respond Systems, Applications, and Services, Seattle, WA, 2005.