Evidentialist Foundationalist Argumentation in Multi-Agent Systems

by

Christopher L. Redford

University of Kansas, 2012

Submitted to the graduate degree program in Computer Science and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of

Doctor of Philosophy.

Dr. Arvin Agah (Chairperson)

Dr. Swapan Chakrabarti

Dr. Luke Huan

Dr. Brian Potetz

Dr. Prajna Dhar

Date Defended: April 9th, 2012

The Dissertation Committee for Christopher L. Redford certifies that this is the approved version of the following dissertation:

Evidentialist Foundationalist Argumentation in Multi-Agent Systems

Dr. Arvin Agah (Chairperson)

Dr. Swapan Chakrabarti

Dr. Luke Huan

Dr. Brian Potetz

Dr. Prajna Dhar

Date approved: April 9th, 2012

Abstract

This dissertation focuses on the explicit grounding of reasoning in evidence directly sensed from the physical world. Based on evidence from human problem solving and successes, this is a straightforward basis for reasoning: to solve problems in the physical world, the information required for solving them must also come from the physical world. What is less straightforward is how to structure the path from evidence to conclusions. Many approaches have been applied to evidence-based reasoning, including probabilistic graphical models and Dempster-Shafer theory. However, with some exceptions, these traditional approaches are often employed to establish confidence in a single binary conclusion, like whether or not there is a blizzard, rather than developing complex groups of scalar conclusions, like where a blizzard's center is, what area it covers, how strong it is, and what components it has. To form conclusions of the latter kind, we employ and further develop the approach of Computational Argumentation.

Specifically, this dissertation develops a novel approach to evidence-based argumentation called Evidentialist Foundationalist Argumentation (EFA). The method is a formal instantiation of the well-established Argumentation Service Platform with Integrated Components (ASPIC) framework. There are two primary approaches to Computational Argumentation. One approach is *structured argumentation* where arguments are structured with premises, inference rules,

conclusions, and arguments based on the conclusions of other arguments, creating a tree-like structure. The other approach is *abstract argumentation* where arguments interact at a higher level through an attack relation. ASPIC unifies the two approaches. EFA instantiates ASPIC specifically for the purpose of reasoning about physical evidence in the form of sensor data. By restricting ASPIC specifically to sensor data, special philosophical and computational advantages are gained. Specifically, all premises in the system (evidence) can be treated as firmly grounded axioms and all arguments' conclusions can be numerically calculated directly from their premises.

EFA could be used as the basis for well-justified, transparent reasoning in many domains including engineering, law, business, medicine, politics, and education. To test its utility as a basis for Computational Argumentation, we apply EFA to a Multi-Agent System working in the problem domain of Sensor Webs on the specific problem of Decentralized Sensor Fusion. In the Multi-Agent Decentralized Sensor Fusion problem, groups of individual agents are assigned to sensor stations that are distributed across a geographical area, forming a Sensor Web. The goal of the system is to strategically share sensor readings between agents to increase the accuracy of each individual agent's model of the geophysical sensing situation. For example, if there is a severe storm, a goal may be for each agent to have an accurate model of the storm's heading, severity, and focal points of activity. Also, since the agents are controlling a Sensor Web, another goal is to use communication judiciously so as to use power efficiently.

To meet these goals, we design a Multi-Agent System called Investigative Argumentation-based Negotiating Agents (IANA). Agents in IANA use EFA as the basis for establishing arguments to model geophysical situations. Upon gathering evidence in the form of sensor readings, the agents form evidence-based arguments using EFA. The agents systematically compare the conclusions of their arguments to other agents. If the agents sufficiently agree on the geophysical situation, they end communication. If they disagree, then they share the evidence for their conclusions, consuming communication resources with the goal of increasing accuracy. They execute this interaction using a Share on Disagreement (SoD) protocol.

IANA is evaluated against two other Multi-Agent System approaches on the basis of accuracy and communication costs, using historical real-world weather data. The first approach is all-to-all communication, called the Complete Data Sharing (CDS) approach. In this system, agents share all observations, maximizing accuracy but at a high communication cost. The second approach is based on Kalman Filtering of conclusions and is called the Conclusion Negotiation Only (CNO) approach. In this system, agents do not share any observations, and instead try to infer the geophysical state based only on each other's conclusions. This approach saves communication costs but sacrifices accuracy.

The results of these experiments have been statistically analyzed using omega-squared effect sizes produced by ANOVA with p-values < 0.05. The IANA system was found to outperform the CDS system for message cost with high effect sizes. The CDS system outperformed the IANA system for accuracy with only small effect

sizes. The IANA system was found to outperform the CNO system for accuracy with mostly high and medium effect sizes. The CNO system outperformed the IANA system for message costs with only small effect sizes. Given these results, the IANA system is preferable for most of the testing scenarios for the problem solved in this dissertation.

Acknowledgements

My greatest thanks go to Professor Arvin Agah for his years of guidance both as my official advisor and before. He has been a fantastic advisor. He constantly found interesting elements within my ideas and helped me develop concrete plans for expressing them. I always left his office with a clearer idea of what I needed to do and the motivation to do it. I would also like to thank Professors Swapan Chakrabarti, Luke Huan, Brian Potetz, and Prajna Dhar for serving on my doctoral committee.

Many professors also played indirect but important roles in helping me achieve my academic goals. Professor Swapan Chakrabarti helped me start my path to research as an undergraduate through many research projects. Professor Nancy Kinnersley served as an inspiring mentor in teaching for years, both before and after I worked with her as a Graduate Teaching Assistant. The many opportunities she gave me in teaching profoundly shaped my teaching experience and philosophy. Professor Man Kong was an inspiration in teaching as well and advised me for my presidency of Upsilon Pi Epsilon. Professor Luke Huan helped me further improve my skills in teaching. Professor Costas Tsatsoulis supported me through a NASA supervised research position that inspired the research of my doctoral work. The encouragement and technical advice given to me by the research engineer Ed Komp were also an important starting point for my doctoral research.

Finally, I would like to thank Crystal Leal for her years of love, support, and encouragement. The process of obtaining a doctoral degree presents many intense challenges and satisfying triumphs. She lived through those experiences with me, reminding me of my capabilities when I was discouraged and celebrating my victories with me when new goals were reached. Gratitude also goes to my parents, Ivan and Marsha, and my sister, Tiffany for their continued love and support.

Contents

List of Figures

1 Introduction

1.1 Motivation

A quintessential goal for Multi-Agent Systems (MAS) is to *share limited resources* among multiple autonomous agents in an effort to collectively achieve one or more *common goals*. The participation of multiple agents is typically required to achieve those goals. Some examples of limited shared resources are an individual agent's time, processing power, or battery power. Other examples are external physical resources such as sensors, physical space, communication channels, or tools. Anything with limited accessibility that an agent or group of agents can utilize to accomplish a goal can be thought of as a limited resource. Maximally utilizing the group's limited resources to realize the group's common goals is a frequently contemplated challenge in MAS.

In current practice, Multi-Agent Systems are typically designed from the top down with specific *cooperative* goals in mind. Multiple agents are often used because the goals are complex and distributed in nature. Sometimes these goals can be achieved more easily from the perspective of multiple autonomous entities rather than from the perspective of one complicated entity. Having multiple agents also allows for graceful degradation of the system, should problems occur, whereas centralized systems have a single point of failure.

One useful application for Multi-Agent Systems is in the problem domain of Decentralized Sensor Fusion (Rosencrantz *et al.*, 2003). In Decentralized Sensor Fusion, multiple independent entities attempt to strategically share observations in a way that maximizes the accuracy of their individual models of the overall situation. This may be used by sensors spread across a geographical landscape to build individual models of a weather situation such as distributed blizzard intensity or tornado locations. The goal in this situation is to maximize the *model accuracy* for each individual agent.

Distributed sensors have several advantages which will be detailed in our Background chapter. One disadvantage they have is that communication between sensors consumes energy, for example battery life. This cost can be called *communication cost*, which will be formally defined in our Materials chapter. Communication cost is necessary for achieving model accuracy. An open question is how best to balance the two.

One approach is to share all observations among all agents. This guarantees that maximum model accuracy will be achieved but at the price of incurring maximum communication cost. We call this approach **Complete Data Sharing (CDS)**. Another approach is for agents to build models based on their own individual observations and only share the conclusions of their models with each other. It could combine these model conclusions using a sophisticated combination algorithm like Kalman Filtering (Olfati-Saber, 2007). This comes at the cost of possibly reducing model accuracy but also possibly lowering communication cost in comparison to CDS. We will call this approach **Conclusion Negotiation Only (CNO)**.

Given these two extremes, there is a possibility for a hybrid approach that strategically shares some observations but only when sharing those observations has the potential to increase accuracy. The hybrid approach we propose is based on Computational Argumentation (Prakken, 2010). This approach takes advantage of the fact that arguments are structured as premises that lead to conclusions. In this proposed system, observations are a type of premise and weather models are a type of conclusion. The formal components of the approach will be detailed in later chapters but the basic idea can be illustrated with an informal example.

The practical idea behind the strategy is that if agents sufficiently agree on their general conclusions, they do not need to share the details of how they reached their conclusions. For example, if two human experts (for example, doctors) agree on their conclusion as to which detailed operation is right for a situation, it is not necessary that they share the studies that lead them each individually to that conclusion. It is not unless there is a disagreement that the experts should share their evidence and come to a consensus about whose conclusion has the most evidence. This scenario is illustrated in Figure 1. In the same way, if agents in a Sensor Web basically agree on their conclusion as to what is happening in the current geophysical situation, then it is not necessary that they share the observations that led them to their conclusions. We will call this approach **Investigative Argumentation-based Negotiating Agents (IANA)**.

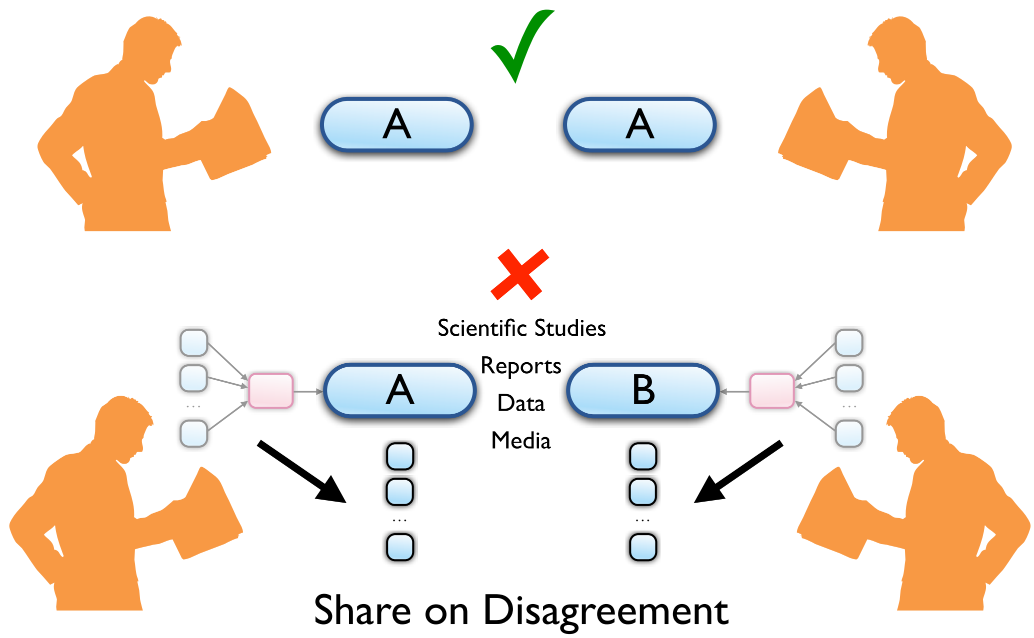

Figure 1. Often two human experts only need to share the evidence for their conclusions if they disagree.

The hypothesized advantage of this approach is conceptually illustrated in Figure 2. A certain amount of data exchange is necessary to achieve the total accuracy for an environmental model. However, at some point the data exchanged can become irrelevant to the accuracy of the model. The data may be redundant. Or the data may not meet certain conditions and thus become irrelevant. At this point, the accuracy plateaus and further exchange of data results in diminishing returns. For an approach that shares no observations but instead shares conclusions (CNO), the accuracy reached may be far from the maximum possible. For an approach that shares all observations (CDS), maximum accuracy may be reached but at the cost of sharing far more data than necessary. For an approach that only shares observations when conclusions are in disagreement (IANA), close to maximum accuracy may be reached and with much lower communication cost.

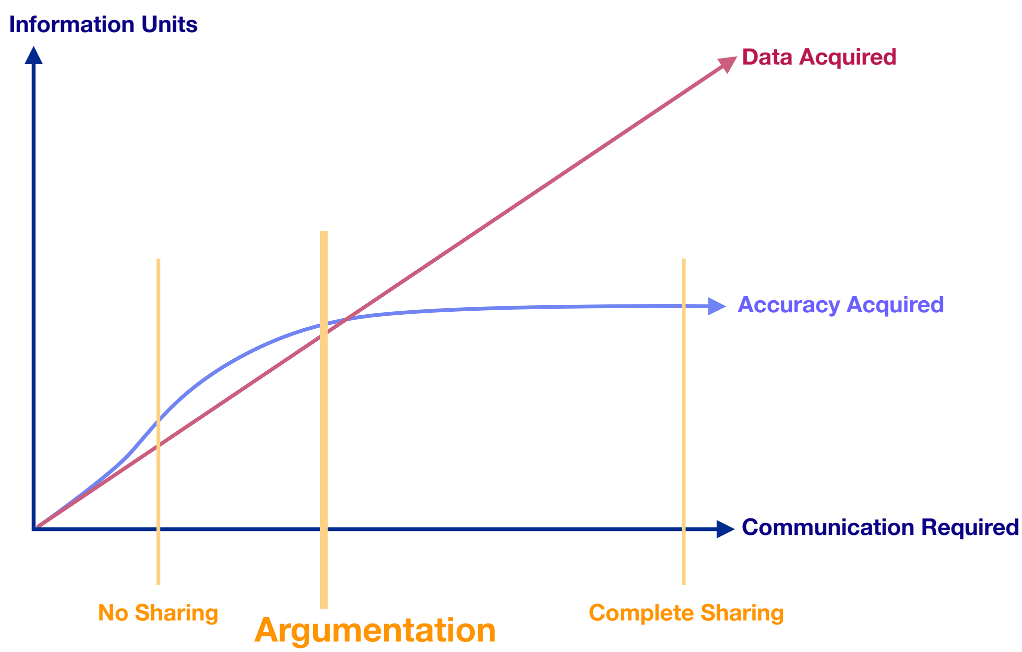

Figure 2. The conceptual advantage of an argumentation-based approach.

The goal of this dissertation is to formally test these hypothesized advantages. In the next section, we will formally established our research hypotheses.

1.2 Research Hypotheses

To evaluate our formal research hypotheses, we will perform a set of experiments. All of the experiments will follow a similar setup. Sensor Webs will be arranged in a grid pattern and each sensor pod (a heterogenous set of sensors) in the grid will be associated with an agent. Each sensor pod will be given access to a snapshot of real-world data associated with its geographical position, drawn from a historical real-world weather scenario. Agents will communicate with each other using one of the three strategies outlined in the previous section: CDS, CNO, or IANA. In the meantime, an Evaluator agent will be given all the data so that it can generate a complete model of the weather scenario for accuracy comparisons. The performance of each system will be evaluated in terms of multiple accuracy and communication metrics, which will be detailed in the Materials chapter.

For the purpose of explaining our research hypotheses, we will briefly explain communication metrics here. **Message cost** is the *total size* in bytes of messages exchanged by a system multiplied by the distance between the nodes exchanging each individual message (since distance increases the power needed for an exchange). **Message amount** is the total number of messages exchanged by a system.

To statistically test the strength of the advantages of one system over another on a metric, ANOVA will be used to produce both the p-values and the effect sizes to evaluate the advantages. We used the effect size categorization of Cohen (1988) to judge the magnitude of the effect sizes. According to Cohen, effect sizes of 0.4 or less have a "small" effect, 0.4 to 0.8 have a "medium" effect, and 0.8 to infinity have a "large" effect. The effect size categorizations recommended by Cohen are still widely used in modern research across many fields (Kotrlik *et al.,* 2003). They are illustrated graphically in Figure 3.

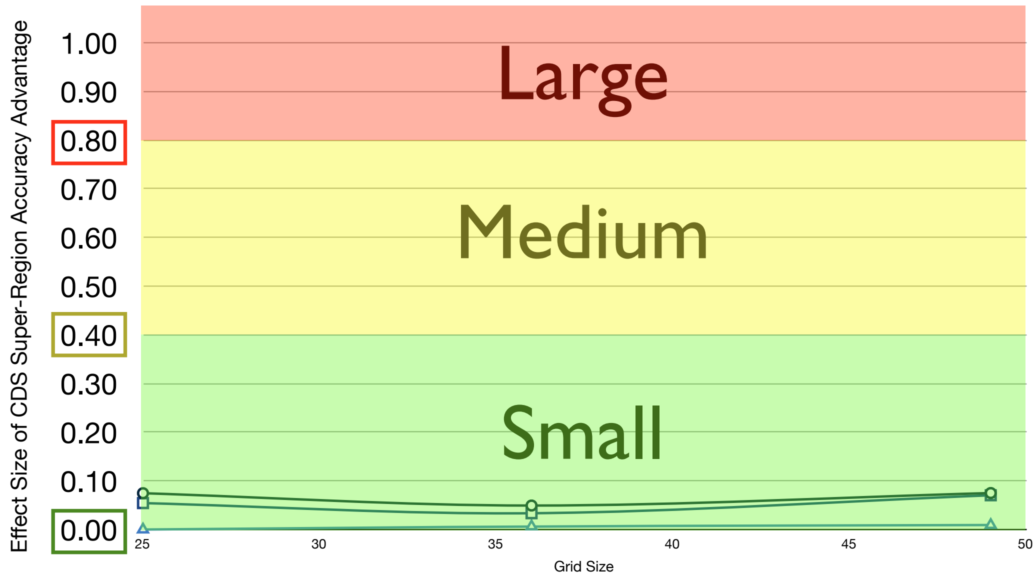

Figure 3. Widely used effect size categorizations.

For these experiments, the IANA system will use a Share on Disagreement (SoD) protocol, which will be explained in detail in a later chapter. The basic principle of this protocol is that agents will only communicate their observations if they sufficiently disagree about the conclusions produced by their models.

1.2.1 Complete Data Sharing

The Complete Data Sharing (CDS) system shares all sensor observations between all agents. It is our hypothesis that CDS will perform better than IANA for all **accuracy metrics** as well as a **message amount** metric. We think CDS will have higher accuracy because it is guaranteed to have 100% accuracy and it is unlikely that IANA can save communication costs without lowering accuracy. We think CDS will send fewer messages because IANA's SoD protocol requires it to send many small messages in order to minimize the exchange of large messages.

However, we also predict that these advantages will all have a *small* effect size. We think the accuracy advantage will be small because, while IANA will probably make *some* mistakes, its SoD protocol should prevent agents from diverging too much from each other, which should usually ensure high accuracy. We think the message amount advantage will be small because CDS is still initiating the same number of *exchanges* as IANA. IANA is just sending a few more messages.

It is also our hypothesis that IANA will perform better than CDS for the **message cost** metric and that this advantage will have a *large* effect size. We think this because the CDS solution sends *all* observations to all other agents, even if those agents do not need those observations to reach similar conclusions and even if those observations are ultimately irrelevant to forming a conclusion. Since IANA is specifically designed to avoid this unnecessary communication, we believe that its communication costs should be much lower.

If these hypotheses are true, it can be argued that, unless maximal accuracy and minimal message amount are required for a system, it is more suitable to use IANA than CDS, due to the strong advantages given by the lower message cost.

1.2.2 Conclusion Negotiation Only

The Conclusion Negotiation Only (CNO) system shares only high-level conclusions produced by individual agent models between agents. The agents then combine their conclusions using Kalman Filtering. It is our hypothesis that

CNO will perform better than IANA for all communication metrics. We think CNO will have fewer **message costs** and lower **message amount** because, intuitively, its protocol requires fewer messages and smaller messages.

However, we also predict that these advantages will all have a *small* effect size. We think the communication advantage will be small because, while IANA will use slightly more messages when exchanging observations it will also *refrain* from sending observations when agents sufficiently agree using its SoD protocol.

It is also our hypothesis that IANA will perform better than CNO for the **accuracy metrics** and that this advantage will have a *large* effect size. We think this because IANA's agents put more effort into reaching consensus by sharing observations when they disagree, as per the SoD protocol.

If these hypotheses are true, then it can be argued that, unless minimal communication costs are required by a system and lower accuracy is acceptable, it is more suitable to use IANA than CNO, due to the strong advantages given by the higher accuracy of IANA.

1.3 Organization

The rest of the chapters of this dissertation are organized as follows:

- In **Background and Related Work**, we will establish the Sensor Web problem domain and the formal Computational Argumentation concepts that underpin our approach, including a detailed overview of the ASPIC framework (Prakken, 2010).

- In **Evidentialist Foundationalist Argumentation** (EFA), we will formally establish the evidence-based argumentation approach as a unique instantiation of the ASPIC framework and its use in the Sensor Web problem domain.
- In **Experiments**, we will describ the technologies used to perform our experiments.
- In **Investigative Argumentation-based Negotiating Agents** (IANA), we will explain how our multi-agent system is designed to use EFA and the Share on Disagreement (SoD) protocol.
- In **Complete Data Sharing** (CDS), we will present the design of our competitive complete sharing system and report on the ***experimental results*** of the comparison to IANA.
- In **Conclusion Negotiation Only** (CNO), we will explain the design of our competitive Kalman Filtering based conclusion negotiating system and report the ***experimental results*** of the comparison to IANA.
- In **Conclusion**, we will evaluate the results of our experiments and speculate on potential applications of the IANA approach to multi-agent Sensor Webs and the EFA approach to argumentation.

2 Background and Related Work

This dissertation is the result of the integration of the fields of Sensor Webs and Computational Argumentation. The current characteristics of each field and their relation to our own work are described in this chapter.

2.1 Sensor Webs

A succinct definition of the Sensor Web concept can be found on the NASA Jet Propulsion Laboratory's (JPL) Sensor Web site:

> *"A networked set of instruments in which information from one or more sensors is automatically used to reconfigure the remainder of the sensors" (JPL, 2009).*

While the concept itself is straightforward, it has taken many different manifestations since its conception by Delin *et al.* (1999). The Sensor Web concept is intentionally broad and abstract (so as to allow any set of sensors to be included under the general term) but this broadness has also lead to some confusion in the literature as to what exactly a concrete, real-world Sensor Web is.

Different groups, over different periods of time, have offered different concrete realizations of what precisely the quintessential example of a Sensor Web is. Some believe the wireless networking concept is at its core (Seada *et al.*, 2006). Others emphasize the system of systems concept (Delin *et al.*, 2005). Still others emphasize the quality of intra-web communications as the distinguishing property (Sherwood *et al.*, 2007).

It may be useful if researchers specify which facet of the core Sensor Web concept they are tackling. That is, instead of referring to “Sensor Web” proper, researchers could formally establish the different sub-fields that have arisen out of the original Sensor Web concept.

Perhaps the Sensor Web concept has outgrown the phase in which it refers to any one sub-discipline. Much like the concept of Artificial Intelligence (AI) before it, which was once only associated with the computation of formal logic, it may be useful for the area of Sensor Web research now stand for a set sub-disciplines centered around a core concept (just as AI research now includes such disparate areas from logic as evolutionary algorithms and computer vision).

In the following sections, we catalogue some of the different manifestations the Sensor Web has taken, which have been successful, and the particular subfields of Sensor Web development that are the focus of our research.

2.1.1 The In-Situ Sensor Web

The first popular conception of the Sensor Web was the *in-situ* Sensor Web. It is meant to refer to the fact that the sensors in this type of Sensor Web are placed in the very environment that they are meant to be sensing, as opposed to sensors which measure remotely, such as those in satellites.

The primary advantages of this in-situ placement, as compared to remote measurements, are:

- Continuous presence in the sensing environment
- Higher measurement accuracy due to proximity to the measured phenomenon
- Higher measurement resolution (assuming a higher density of sensor placement than the resolution of the remote method being compared)

Also, some measurements, such as soil moisture or seismic readings, are very difficult to sense remotely, making in-situ sensors the only viable method of obtaining those types of observations.

A concise argument for the efficacy of the in-situ Sensor Web as compared to lone in-situ sensors is given by Delin (2002) and is characterized in Figure 4.

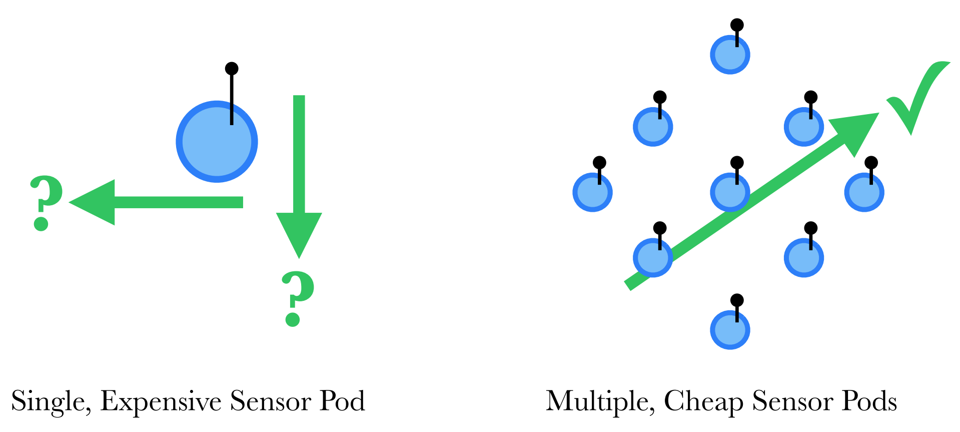

Figure 4. Single sensors lack the advantage of sensing the spatial relationship between measurements.

A disadvantage of using a single sensor is that it receives a minimum amount of spatial information. This results in a minimum amount of information about any kind of vector fields within the sensing environment, including cyclonic winds, temperature gradients, and hydrologic water movement (Delin *et al.*, 2004).

The individual physical nodes in a Sensor Web are called pods, as a single pod can be associated with multiple sensors. Basically, a pod consists of one or more sensors, a communication component, and a processing component. For the purpose of discussing communication issues, calling them pods also distinguishes them from nodes which could be either single pods or entire Sensor Webs.

The most commonly proposed method for inter-pod communication in an in-situ Sensor Web is pod-to-pod routing. The reason for this being that direct transmission from the transmitting pod to the receiving pod is more power intensive than passing the message between pods. This is because the power required to transmit increases by a polynomial factor with distance, as given by the equation in Figure 5, where D is distance, λ is wavelength, the P's are power transmitted and received, and m is a constant ranging from 2 to 4, depending on the environment (Delin *et al.*, 2001). For this reason, hopping the data between pods is more power efficient.

$$P_{transmit} \propto \left(\frac{D}{\lambda}\right)^{m} \cdot P_{\text{receive}}$$

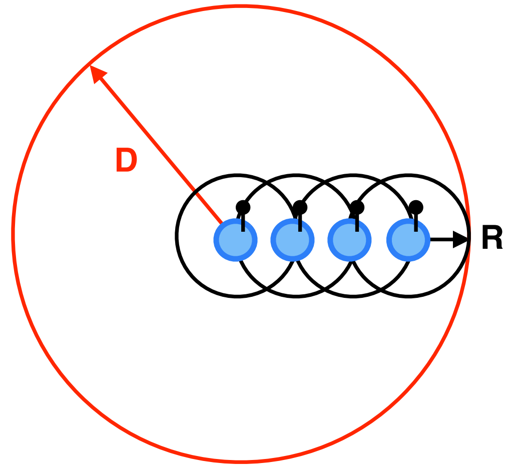

Figure 5. The power required to transmit over distance D increases by a polynomial factor as D increases. Hopping the data by a distance of R from pod to pod is more power efficient than directly transmitting the data over distance D.

This situation is an exemplar illustration of how attempts at physical implementation of a hardware concept provide important feedback on the overall design. A more comprehensive delineation of what has been learned from attempts at implementing concrete Sensor Webs can be found in (Delin *et al.*, 2005). We will provide a slightly broader overview of how the physical concept of the in-situ Sensor Web has evolved in this section.

Delin *et al.* (2000) initially conceptualized the in-situ Sensor Web pod as a small device (the size of a gum ball container), that could be scattered over a sensing landscape, such as a forest or the surface of Mars, creating a continual sensing presence that could be relayed back to mission control via a mother or portal pod.

One of the primary advantages of this pod concept was that it was low cost. This is why the pods could be scattered in a relatively unstructured manner, thus providing a variety of spatial sensing configurations. Pods like these could be thought of almost like a type of sensing "ammunition" that could be shot or painted across a landscape, providing instant sensing awareness wherever they landed. Another advantage of their low cost was that they could be "reseeded" into an in-situ Sensor Web that had pods that had failed (Delin *et al.*, 2001). In this way, in-situ Sensor Webs could be healed much the same way as the human body heals: not by repairing dead cells but by simply discarding them and replacing them with new cells.

Preliminary lab experiments demonstrated that the gum ball sized pods were able to execute the multi-hop communication protocol well. As different pods in the web were relocated, it was demonstrated that they could resiliently reform the communication fabric and relay all pod readings to the portal pod, which was connected to a laptop (Delin *et al.*, 2000). Given that these tests likely took place in an indoor environment, this pod design could still be an important consideration for indoor in-situ Sensor Webs.

However, after a few years of experience deploying Sensor Webs for use by collaborators in outdoor environments, Delin's team concluded that the gum ball sized pod design simply was not well suited for the types of outdoor environments encountered at least on Earth. They presented a set of realizations as a result of this experience that amounted to a strong discouragement from the idea of using gum ball sized pods for out-door in-situ Sensor Web applications (Delin *et al.*, 2005).

Delin *et al.* (2005) delineate four major problems with the scattered gum ball sized pod design. The first three are related to hardware design limitations related to the small gum ball size:

- **Power:** The larger the volume in the pod, the more volume is available for battery technology. The power required for wireless communication with other nodes dictates a lower bound on how much power is required for the system. Also, since solar power is currently the most practical method of energy harvesting to recharge the pod batteries, the smaller the surface area of the pod, the less energy it can harvest.
- **Antenna:** The laws of physics dictate the appropriate antenna geometry for a given operating frequency. Antenna sizes cannot shrink if a particular communication range is desired. And most of the reported real-world deployments required communication ranges of at least tens of meters.
- **Sensors:** Small pods are ideal for measuring simple parameters (temperature, humidity, light, etc.). However more complicated transducers, such as those used for soil moisture measurement (which sometimes must stretch as far as half a meter under ground), gas sensing (which often require a certain volume of gas), or seismometers (which require a certain mass for appropriate mechanical resonant frequencies) put another lower bound on how small a pod can be. There is little to be gained from shrinking the pod platform if the sensors themselves remain the limiting size element.

The last is related to environmental limitations with the imagined scattering method of gum ball pod deployment:

- **Placement:** Most applications require tracking specific pod locations to a high precision. It is therefore highly unlikely that any wireless sensor network will simply be sprinkled over large areas. In addition, coupling sensors into the environment will also prevent such a passive deployment. For example, neither subterranean nor seismic sensors can be deployed by a sprinkling technique, as both require laborious efforts for appropriate sensor mounting. There are also applications, particularly those involving agriculture, where pod placements must be compatible with existing operations, such as harvesting. Pod placement very close to the ground can also limit transmission distance.

As a result of these limitations, the actual Sensor Webs that Delin's team deployed in outdoor areas took a larger, more rugged design. The pods were placed on firmly grounded stands and their circuitry was encased in a rigid box, with only the sensor detection mechanism reaching outside. The pods used in other in-situ Sensor Webs have similar design likely for similar reasons (Kedar *et al.*, 2008).

These new Sensor Web pods were used, with little or no modification between applications, in deployments in botanical gardens in temperate climates, for biological sensing in Antarctic climates, and in hydrological recharge basins in desert climates.

Another example of a real-world in-situ Sensor Web application that required relatively large Sensor Web pods is documented in (Heavner *et al.*, 2008). Their approach shows full realization of the limitations of smaller sensor pods and therefore favors larger ones. The large solar panels used are an issue directly addressed by Delin *et al.* (2005).

2.1.2 Web of Webs

As mentioned earlier, each pod in an in-situ Sensor Web, as well as the entire in-situ Sensor Web itself, can be thought of as a node in a single unified communications fabric. This concept is illustrated in Figure 6.

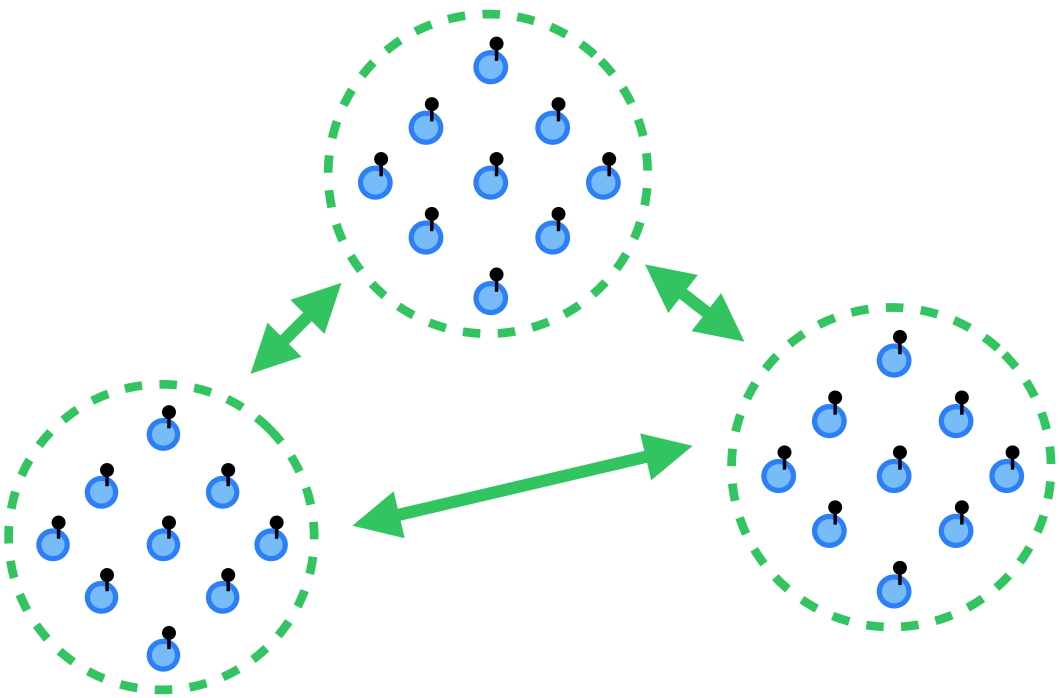

Figure 6. Sensor Webs can be conceptualized as a web of webs.

Using this model, even remote sensors like satellites can be included in the larger Sensor Web structure. This structure of substructures is a familiar concept as it is, in many ways, directly analogous to the structure of the Internet, where multiple local networks are connected in a hierarchy to other networks. Also analogous to the Internet is the fact that sensors in different sub-webs can perform tasks for each other despite the fact that they are in different local webs. So if an in-situ ground sensor wants to task a satellite to focus its measurements in a certain area using certain instrument, it can do so and vice-versa. These two nodes in the super-web structure can also share information so that the observations of one can affect the other.

A common remote sensing technology to use within a web of webs is satellite technology. One of the Sensor Webs to make use of the web of webs concept is the Optimized Autonomous Space In-situ Sensor-Web (OASIS) (Kedar *et al.*, 2008). The point of the OASIS Sensor Web is to sense volcanic activity. Similar to the proposed Sensor Web application, the purpose is hazard sensing to facilitate early emergency response. Their prototype is deployed on Mount St. Helens.

What makes OASIS unique is its use of the EO-1 satellite to coordinate the communication resources and power usage of the in-situ web. In turn, the in-situ web also tasks the EO-1 satellite with requests to sense specific areas on the volcano. Thus, a feedback loop between the two webs is formed, where the EO-1 satellite can be thought of as a single-pod web (Sherwood *et al.*, 2007).

2.1.3 Core Metrics

The primary resource involved in Sensor Web management is **power**, often battery power. Most other resource concerns are reducible to power concerns in some way. For example, both communication costs (the resource consumed when agents communicate) and mobility concerns can be reduced to power concerns because the reason one would want to minimize communication and mobility is to minimize power usage. This relationship is illustrated in Figure 7.

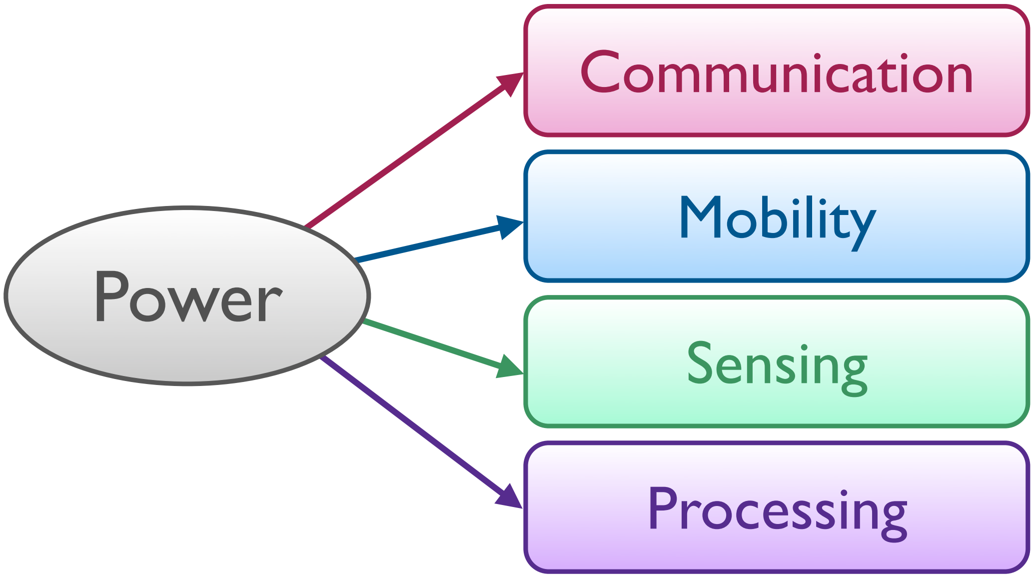

Figure 7. Power is the driving force behind any resource management concern in a Sensor Web.

The quantity and quality of data obtained by a Sensor Web are also critical. The acquisition of sensor **information** lies behind the very motivation for deploying the Sensor Web.

A system could be judged by:

- How much data it gathers: Does the system gather enough data to make good judgements and be useful as an instrument?
- How well it interprets the data: Is the system able to find meaningful patterns within data and extract important summaries of those patterns that highlight the most important data or trends in data?
- How saturated the web is with important data: Does every node know what it needs to know to sense effectively?

Especially in systems that are meant to detect hazards, **response time** could be an important consideration. Also, the faster a system can process sensing data, the more processing tasks for which it can be responsible.

In practice, most modern systems do not yet consider speed an important criteria. This is because most modern application areas involve the sensing of phenomena that take several hours and sometimes several days to transpire. Examples include flooding (Delin *et al.*, 2004), biological activity (Delin *et al.*, 2003), or geological activity (Kendar, 2008). These processes typically take more than enough time for modern processors and communications systems to manage sensing data. However, future applications, such as Intelligent Transportation Systems, may require faster response time.

2.1.4 Intelligent Systems

There have been a few approaches to applying intelligence to the configuration of Sensor Webs. Lou *et al.* (2008) use a centralized intelligent system to configure a Sensor Web that responds to forest fires. Williams *et al.* (2008) apply intelligence to snowmobiles in their study of mobile Sensor Webs. They have implemented an effective method for routing the paths taken by the snowmobiles in moving from one difficult terrain to another.

As stated in the Introduction, in practice, many Multi-Agent Systems are also effective for collaborative goals such as those found in Sensor Webs. Multiple agents are often used because the problem to be solved is complex and distributed in nature.

These problems typically can more easily be solved from the perspective of multiple autonomous entities rather than from the perspective of one very complicated entity. Having multiple agents also allows for graceful degradation of the system, should problems occur, whereas centralized systems have a single point of failure (Weiss, 1999; Poslad, 2007; Aldewereld *et al.*, 2008; Jong, 2008).

One example of an agent approach to Sensor Webs is the use of a science agent currently installed on one of NASA's satellites (Sherwood *et al.*, 2007). Also, there are also many Multi-Agent approaches to configuring Sensor Webs (Otte *et al.*, 2008; Tsatsoulis, 2008; Tynan *et al.*, 2008; Witt *et al.*, 2008).

Notably, none of the reviewed approaches provide quantitative tests of the mechanisms they used, as we will provide in our work. Also, none of the above approaches attempt to use structured evidence-based argumentation-based mechanisms as the basis for their Sensor Web conflict-resolution process. This makes our approach novel within Sensor Web work. Another advantage of our approach is the use of the agent middleware Jadex (Jadex, 2009). This allows us to focus on standardized high-level agent design which can be more easily transmitted to other researchers than ad-hoc Multi-Agent Systems.

2.1.5 Sensor Fusion

The Sensor Web architecture provides many possible applications. For the purpose of this dissertation, we will focus on the specific application of Decentralized Sensor Fusion. In this application, distributed sensor pods have individual information databases. Their goal is to maximize the accuracy of each individual's overall model of a sensing situation, as shown in Figure 8. This type of distribution is analogous to information management in human society, where each individual must be taught and observe for themselves because no single individual can be responsible for all actions. We will use our proposed IANA system to solve the Decentralized Sensor Fusion problem for producing accurate models of a weather sensing situation.

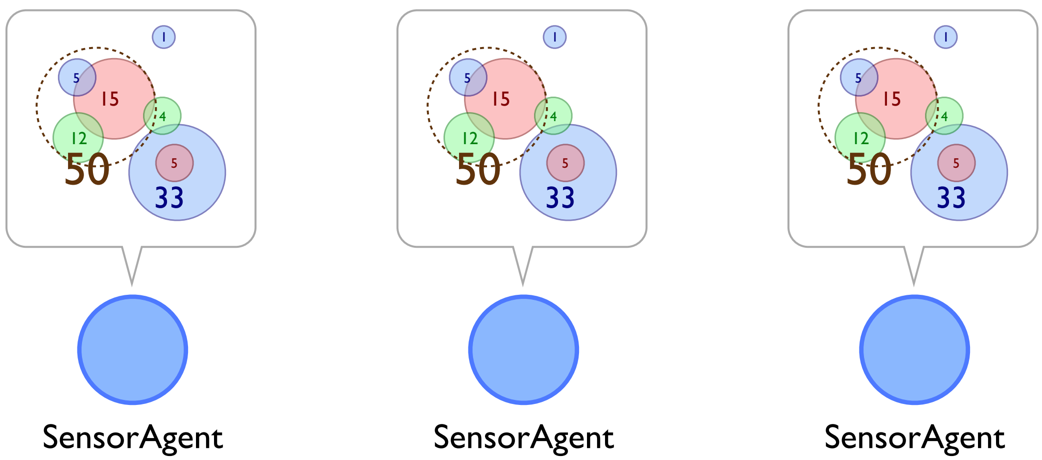

Figure 8. The goal of the Decentralized Sensor Fusion problem is to have accurate modeling in all of the individual Sensor Web participants.

An example of another multi-agent Decentralized Sensor Fusion system can be found in Pavlin et al. (2010). Like IANA, the system is used to create an accurate model of a sensing situation distributed across a group of agents attached to different sensors. An example used in Pavlin et al. (2010) is gas detection and an example in the related approach of Pavlin et al. (2006) is fire detection. They use a probabilistic Bayesian approach for inference, so, unlike our approach, continuous scalar values cannot be produced by their system.

In addition to Bayesian approaches, another popular Sensor Fusion approach is the application of Dempster-Shafer evidence theory, such as the systems employed by Basir et al. (2007) or Hong et al. (2009). The approach of Hong et al. (2009) in particular is applied to the situational sensing problem of sensing events in a "Smart Home". Also, Dempster-Shafer applications employ inference trees, just as EFA and Bayesian approaches do. Similar to Bayesian approaches, Dempster-Shafer approaches, as employed by these papers, cannot directly produce continuous scalar values or data structures that represent detailed models.

A type of Sensor Fusion approach that can process and produce continuous scalar data, is Distributed Kalman Filtering (Gan et al., 2001; Olfati-Saber, 2005; Olfati-Saber, 2007). Due to its ability to process continuous data, Distributed Kalman Filtering (DKF) is an ideal competitor for our system and a Kalman Filtering approach is used in the competing CNO system. However, these approaches are all applied to the Sensor Fusion problem of tracking a moving target rather than situational modelling, so we will be developing the CNO system ourselves.

2.2 Argumentation

Argumentation, as opposed to formal (deductive) logic, has been favored in the areas of law, philosophy, and artificial intelligence due to the following two properties that it does not share with formal (deductive) logic:

- **Defeasibility:** Conclusions about complex environments are often tentative in nature. The limited knowledge an agent can have about an environment is demonstrated by the frame problem (Minsky, 1975). Allowing knowledge to be used now and revised later frees an agent from being logically paralyzed when it does not have certain knowledge but without being ignorantly misinformed if that knowledge can be revised by later observations or conversations with other agents. An agent can make use of the knowledge it has without being limited by that knowledge if it becomes better informed later.

- **Inclusion of Probable Conclusions:** Deductive knowledge systems only allow certain conclusions. Uncertain conclusions, no matter how probable, are deductively invalid. However, most conclusions that an agent, whether human or computational, can make about a complex environment are tentative in nature. Further observation may reveal a conclusion to be improbable or even false. But integrating their current knowledge to form the most probable conclusions can help agents make better informed decisions than they would otherwise, and importantly, can lead to further investigation to increase or decrease the agent's certainty of those conclusions.

The basic structure of an argument as developed in the philosophical tradition can be seen in Figure 9.

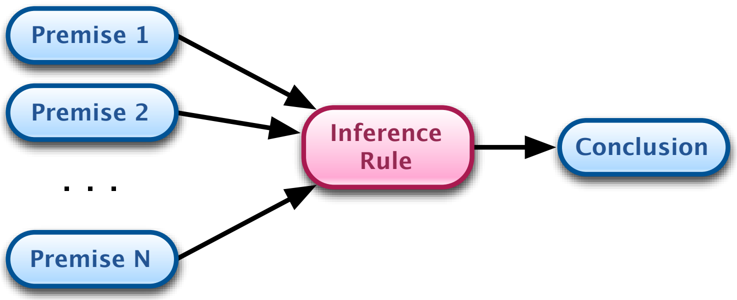

Figure 9. Basic Argument Structure.

All philosophical arguments have the following components (Hurley, 2003):

- **Premises:** The pieces of information, which, when combined with each other and the inference rule, lead to the conclusion.
- **Inference Rule:** The connective by which the premises validate the conclusion. This is often implicit in human argumentation but must always be made explicit in Computational Argumentation.
- **Conclusion:** The assertion that the premises justify. Like premises, conclusions are units of information and therefore can be used as the premises of other arguments. All arguments have only one conclusion. If a proposal contains more than one conclusion, it has more than one argument.

The following is an example of a simple philosophical argument:

Premise 1. The sun has risen every day I have checked for it

Premise 2. I have no reason to think it will not rise again

Conclusion. The sun will rise again tomorrow

Modern interest in argumentation, whether rhetorical, legal, or computational inspired many developments from Stephen Toulmin. Toulmin's Model of Argument (Toulmin, 1958) was first presented as a basis for the analysis of legal arguments. It was later championed by rhetorical scholars and now has been used by Computer Scientists in the development of Computational Argumentation.

Much of Computational Argumentation, in its current form, was also inspired by the work of (Dung *et al.*, 1995). It has since received significant development by researchers committed to the topic (Amgoud *et al.*, 2000; McBurney, 2002; Reed *et al.*, 2007; Rahwan *et al.*, 2007).

At least three conferences focused on Computational Argumentation have been developed and meet regularly (at the time of this writing): Argumentation in Multi-Agent Systems (ArgMAS, 2009), Computational Models of Natural Argument (CMNA, 2009), and Computational Models of Argument (COMMA, 2009).

Interest in Computational Argumentation has lead to the development of at least one committed project, the Argumentation Service Platform with Integrated Components (ASPIC), which was funded by the European Union (ASPIC, 2009). ASPIC, in turn, resulted in the ASPIC argumentation framework, which is the basis for the Evidentialist Foundationalist Argumentation (EFA) used in this work. The ASPIC framework will be described in detail in the rest of this chapter.

2.2.1 The ASPIC Framework

In Prakken (2010), the ASPIC framework of Amgoud *et al.* (2006) was extended and unified with work in abstract argumentation initiated by Dung (1995), as well as many approaches to argumentation with structured arguments (Pollock, 1994; Vreeswijk, 1997; Amgoud *et al.*, 2006; Caminada *et al.*, 2007; Gordon *et al.*, 2007; Dung *et al.*, 2009). The framework defines arguments as inference trees formed by applying strict and defeasible inference rules. It also provides classifications of different types of premises that can be used to support an argument, taken from a knowledge base. It also provides various classifications of attack and defeat, depending on whether premises, rules, or conclusions are attacked. Finally, it satisfies many rationality postulates. Given these features, ASPIC is a comprehensive framework which can be used to formally connect and contrast many different approaches to argumentation. ASPIC can also be used as the foundation for defining and classifying new types of argumentation and contrasting them with other approaches.

Prakken (2010) distinguishes between an *argumentation system (AS)*, an *argumentation theory (AT)*, and an *abstract argumentation framework (AF)*. Specifically, an AT *includes* an AS as a member and an AF *corresponds* to an AT. An AS is used to define the way that structured arguments can be formed. An AT is used to associate an AS with a knowledge base and an argument ordering. And an AF is used to specify a relation by which arguments produced by an AT can defeat each other.

Prakken's definitions will be used as the foundation to define the AS, AT, and AF used in this dissertation. For the sake of completeness, relevant definitions from

Prakken (2010) will be reproduced here.

2.2.2 ARGUMENTATION SYSTEM

The first relevant concept is that of AS.

Definition 1: [Argumentation System] An *argumentation system* is a tuple AS = (L, Contr, R, $\leq$) where

- L is a **logical language**
- Contr is a **contrariness function** from L to 2^L
- $R = R_s \cup R_d$ is a set of **strict** (R_s) and **defeasible** (R_d) **inference rules** such that

  $R_s \cap R_d = \emptyset$
- $\leq$ is a **partial preorder** on R_d

The contrariness function can be used to distinguish the concepts of *contrary* and *contradictory* statements. In propositional logic, only the concept of *contradictory* statements is allowed. But in argumentation systems, the concept of *contrary* statements is meaningful.

Definition 2: [Contrariness Function] Let L, a set, be a logical language. A *contrariness function* Contr maps L to 2^{L}. If $p \in$ Contr(q), then if $q \notin$ Contr(p) then p is called the *contrary* of q, otherwise p and q are called *contradictory*. The latter case is denoted by $p = -q$ (i.e. $p \in$ Contr(q) and $q \in$ Contr(p)).

For example, if Contr(q) = {p, t, r} and Contr(p) = ∅ in an AS, then the instantiation of both p and q by that AS means that p is *contrary* to q in that instantiated set. This is useful for representing the situation where p can be used to counter q but q cannot be used to counter p. For example, when p represents an exception to a rule that produces q. In addition to this useful purpose, contrariness functions can also be used to specify a *consistent language set*.

Definition 3: [Consistent Language Set] Let P ⊆ L. P is *consistent* iff $\nexists\, p, q \in$ P such that $p \in$ Contr(q). Otherwise, P is *inconsistent*.

Rules are used to infer a *consequent* from a set of *antecedents*. Arguments are built by applying inference rules to subsets of L.

Definition 4: [Strict and Defeasible Rules] Let $p_0,\ldots, p_n$ be elements of L.

- A *strict rule* has the form $p_0, \ldots, p_n \rightarrow q$, informally meaning that if $p_0, \ldots, p_n$ hold, then *without exception* it holds that q
- A *defeasible rule* has the form $p_0, \ldots, p_n \Rightarrow q$, informally meaning that if $p_0, \ldots, p_n$ hold, then *presumably* it holds that q

Strict rules can be used to represent a deductive relationship between consequent and antecedents. Defeasible rules can be used to represent an inductive relationship between consequent and antecedents. This distinction will be important for the AS proposed by this dissertation.

The next relevant definition is about the *knowledge base* used for antecedents and consequents in an AS.

Definition 5: [Knowledge Base] A *knowledge base* in an argumentation system (L, Contr, R, $\leq$) is a pair $(K, \leq')$ where $K \subseteq$ L and $\leq'$ is a partial preorder on $K \setminus K_n$. Here $K = K_n \cup K_p \cup K_a \cup K_i$ where these subsets of K are disjoint and

- K_n is a set of (necessary) *axioms*. Intuitively, arguments cannot be attacked on their axiom premises.
- K_p is a set of *ordinary premises*. Intuitively, arguments can be attacked on

their ordinary premises, and whether this results in defeat must be determined by comparing the attacker and the attacked premise

- K_a is a set of *assumptions*. Intuitively, arguments can be attacked on their assumptions, where these attacks always succeed.
- K_i is a set of *issues*. Intuitively, arguments of which the premises include an issue are never acceptable. An issue must always be backed with a further argument.

Prakken (2010) defines *argument* using a single definition. For the sake of having access to necessary distinctions in the current work, Prakken's argument definition will be organized into sub-definitions, while retaining the same content and meaning. For all of the argument definitions, the following functions will be used: *Prem* returns all the formulas *K* (called *premises*) use to build an argument, *Conc* returns its conclusion, *Sub* returns all its sub-arguments, *DefRules* returns all the defeasible rules of the argument and, finally, *TopRule* returns the last inference rule used by the argument.

Definition 6: [Premise-Encapsulating Argument] A *premise encapsulating argument* A is:

p if $p \in K$ with:

Prem(A) = $\{p\}$,

Conc(A) = p,

Sub(A) = $\{p\}$,

DefRules(A) = $\varnothing$,

TopRule(A) = undefined.

One purpose of the premise-encapsulating argument is to provide a way to recursively define strict-topped and defeasible-topped arguments, whether they are based directly on members of K or on the conclusions of other strict-topped or defeasible-topped arguments.

Definition 7: [Strict-Topped Argument] A *strict-topped argument* A is:

$A_0, \ldots, A_n \rightarrow q$ if $A_0, \ldots, A_n$ are arguments such that there exists a *strict rule*

$\mathrm{Conc}(A_0), \ldots, \mathrm{Conc}(A_n) \rightarrow q$ in R_s,

$\mathrm{Prem}(A) = \mathrm{Prem}(A_0) \cup \ldots \cup \mathrm{Prem}(A_n)$,

Conc(A) = q,

$\mathrm{Sub}(A) = \mathrm{Sub}(A_0) \cup \ldots \cup \mathrm{Sub}(A_n) \cup \{A\}$,

$\mathrm{DefRules}(A) = \mathrm{DefRules}(A_0) \cup \ldots \cup \mathrm{DefRules}(A_n)$,

$\mathrm{TopRule}(A) = \mathrm{Conc}(A_0), \ldots \mathrm{Conc}(A_n) \rightarrow q$.

The defining quality of a strict-topped argument A is that it employs a strict rule for TopRule(A). As it can be observed in the definition, only TopRule(A) is required to be strict. Defeasible rules may be used in the arguments of Sub(A).

Definition 8: [Defeasible-Topped Argument] A *defeasible-topped argument* A is:

$A_0, \ldots, A_n \Rightarrow q$ if $A_0, \ldots, A_n$ are arguments such that there exists a *defeasible rule*

$Conc(A_0), \ldots, Conc(A_n) \Rightarrow q$ in R_d,

$Prem(A) = Prem(A_0) \cup \ldots \cup Prem(A_n)$,

$Conc(A) = q$,

$Sub(A) = Sub(A_0) \cup \ldots \cup Sub(A_n) \cup \{A\}$,

$DefRules(A) = DefRules(A_0) \cup \ldots \cup DefRules(A_n) \cup$

$\{Conc(A_0), \ldots, Conc(A_n) \Rightarrow q\}$,

$TopRule(A) = Conc(A_0), \ldots Conc(A_n) \Rightarrow q$.

A defeasible-topped argument A has the same qualities as a strict-topped argument with the exception that TopRule(A) is defeasible and DefRules(A) includes TopRule(A) as a member. The inclusion of TopRule(A) as a member of DefRules(A) reveals how DefRules(A) is populated for arguments for which A is a sub-argument. The final inclusive definition can now be introduced.

Definition 9: [Argument] An *argument* A is a premise-encapsulating argument, a strict-topped argument, or a defeasible-topped argument.

From these definitions, it is clear that strict-topped and defeasible-topped arguments can be used to form an arbitrarily large inference tree structure with the conclusions of sub-arguments serving as the premises of larger arguments, as shown in Figure 10. Prakken (2010) also introduces some useful distinguishing properties for arguments. These properties reveal an important role for the DefRules function in previous definitions.

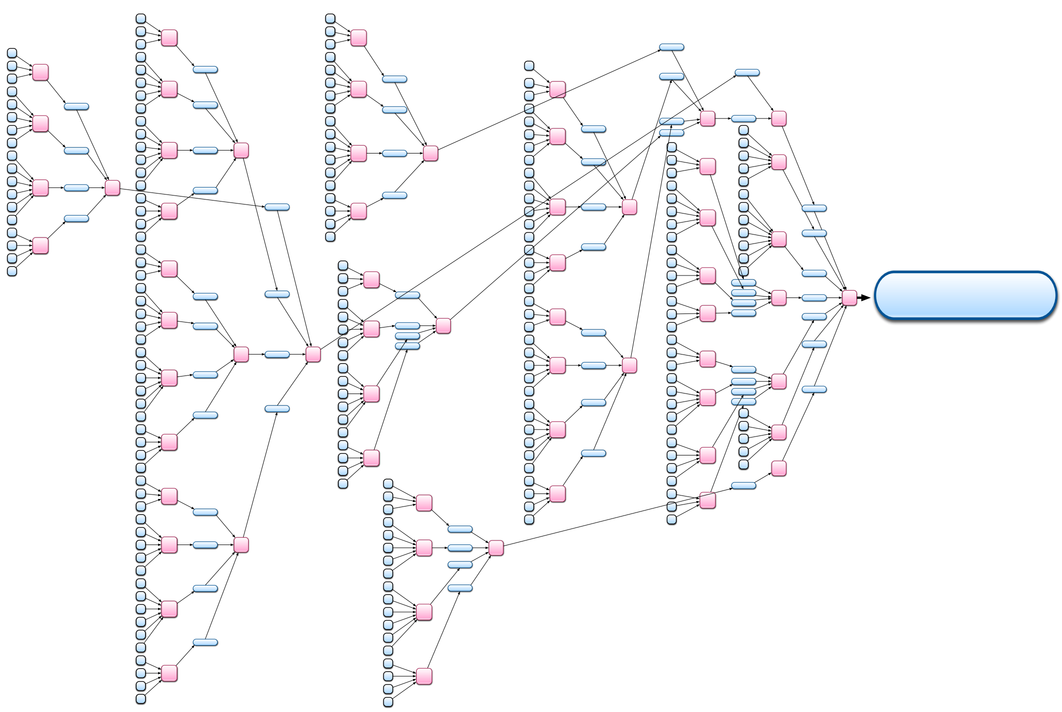

Figure 10. Arguments in ASPIC can have arbitrarily large hierarchies of sub arguments leading to a final conclusion.

Definition 10: [Argument Properties] An argument A is

- *strict* if DefRules(A) = ∅
- *defeasible* if DefRules(A) ≠ ∅
- *firm* if Prem(A) ⊆ K_n
- *plausible* if Prem(A) ⊈ K_n

So premise-encapsulating arguments and strict-topped arguments with only premise-encapsulating arguments or other strict-topped arguments as sub-arguments can be called *strict*. All other arguments are *defeasible*. And arguments with Conc(A) ∈ K_n with only other such arguments as sub-arguments (or no sub-arguments, in the case of eligible premise-encapsulating arguments) can be called *firm*. All other arguments are *plausible*.

With these argument properties defined, important distinctions can be made between entire argumentation theories. As will be explained later, such distinctions can be made between the AT defined in this dissertation and an AT defined in Prakken (2010) based on Dung (2009).

2.2.3 Argumentation Theory

The preceding definitions lay most of the foundation for formally defining the concept of AT. All that is left to define is the concept of admissible argument orderings . Here, $\leq$ is a partial preorder such that A $\leq$ B means that B is at least as 'good' as A. And A $<$ B means that A $\leq$ B and B $\npreceq$ A, meaning that A can be strictly ordered lower than B (Prakken, 2010).

Definition 11: [Admissible Argument Orderings] Let $\mathscr{A}$ be a set of arguments. Then a partial preorder $\preceq$ on $\mathscr{A}$ is an *admissible argument ordering* iff

(1) if B is firm and strict and A is defeasible or plausible, then A $\prec$ B

(2) if $A = A_0, \ldots, A_n \rightarrow q$ then for all $1 \leq i \leq n$, $A \preceq A_i$ and for some $1 \leq i \leq n$, $A_i \preceq A$.

As stated in Prakken (2010), the first condition says that strict-and-firm arguments are stronger than all other arguments, while the second condition says that a strict inference cannot make an argument weaker or stronger. Now the concept of AT can be defined.

Definition 12: [Argumentation Theory] An *argumentation theory* is the triplet AT = (AS, KB, $\preceq$) where AS is an argumentation system, KB is a knowledge base in AS and $\preceq$ is an argument ordering on the set of all arguments that can be constructed from KB in AS (henceforth called the set of arguments on the basis of AT).

2.2.4 Abstract Argumentation Framework

The final concept to be defined is that of the AF. This is used to specify the defeat relations for an AT. Prakken (2010) comprehensively formalizes three different methods for both attack and successful defeat following an attack. These three methods are centered around the three different properties of an argument that can be attacked: its premises, its use of an inference rule, or its conclusion. In the application of the AF defined for this dissertation, only attack and defeat on conclusions is relevant, so only the definitions related to them will be reproduced from Prakken (2010). A complete treatment of all types of possible attack and defeat, refer to Prakken (2010).

Definition 13: [Rebutting Attack] Argument A *rebuts* argument B (on B′) iff Conc(A) $\in$ Contr(q) for some B′ $\in$ Sub(B) of the form $B''_0, \ldots, B''_n \Rightarrow q$. In such a case, A *contrary-rebuts* B iff Conc(A) is a contrary of q.

This defines the attack of a conclusion within B on the basis of a conclusion within A being contrary to it. Prakken also defines the concept of an *undermining* attack on the premises of an argument, which excludes axiom premises. In the AT used by the current work, all bottom-level premises are considered axioms, so this attack is not applicable. Prakken also defines the concept of an *undercutting* attack on the use of an inference rule by an argument in the case of an exception. In the AT used by the current work, rule exceptions are not needed to incorporate defeasibility, so this attack is not applicable either.

Definition 14: [Successful Rebuttal] Argument A *successfully rebuts* argument B if A rebuts B on B′ and either A contrary-rebuts B′ or A $\nprec$ B′.

In the AT used by the current work, all rebuttals will be made on the basis of ordering. Prakken also defines successful undermining which is not applicable to the AT used by the current work since that attack is not applicable.

Definition 15: [Defeat] Argument A *defeats* argument B iff no premise of A is an issue and A undercuts or successfully rebuts or successfully undermines B. Argument A *strictly defeats* argument B if A defeats B and B does not defeat A.

As expected from previous statements, only defeats based on successful rebuttals are applicable for the AT used in the current work. With all of the previous definitions established, the final central concept of AF from Prakken's framework can be defined.

Definition 16: [Abstract Argumentation Framework] An *abstract argumentation framework (AF) corresponding to an argumentation theory AT* is a pair $<\mathscr{A}, Def>$ such that:

- $\mathscr{A}$ is the set of arguments on the basis of AT as defined by Definition 12
- *Def* is the relation on $\mathscr{A}$ given by Definition 15

This final definition from Prakken (2010) unifies the overall framework with Dung (1995) and many other important works in argumentation that have been based on it. Prakken (2010) expands on these basic definitions to show how this overall framework can be used to meet several rationality postulates.

Prakken also shows that the Assumption-Based Argumentation (ABA) used in Dung (2009) is a special case of this overall framework with only strict inference rules, only assumption-type premises, and no preferences. This means that all arguments in ABA are *strict* and *plausible*. The approach used in the current work, called **Evidentialist Foundationalist Argumentation** (EFA), is also a special case of Prakken's overall framework with only defeasible inference rules, only axiom-type premises, no partial preorder on K, and several other properties that will be discussed in detail in the next section. This means that all arguments in EFA are *defeasible* and *firm*, making ABA and EFA exemplar contrasts to each other under Prakken's classifications.

3 Evidentialist Foundationalist Argumentation

We will now extend the ASPIC framework described in the previous chapter to establish a new type of Computational Argumentation approach called **Evidentialist Foundationalist Argumentation** (EFA), as first described in Redford and Agah (2012). EFA is inspired by the reasoning used in Evidentialist epistemology, as found in the philosophical work of Conee *et al.* (2004). Evidentialist epistemologies are focused on the use of evidence to establish claims. EFA can be seen as a formal complement of other approaches to evidence-based argumentation which may be compatible with ASPIC but have not yet been formally unified with it (Oren *et al.*, 2007; Oren *et al.*, 2008; Ontañón *et al.*, 2010).

Evidence is a central concern in argumentation generally, particularly in the domain of law. One notable approach is Bex *et al.* (2003), which generalizes different evidential argument schemes used in law for the purpose of applying them to sense-making systems using the domain independent sense-making tool Araucaria (Reed *et al.*, 2001). The schemes were generalized in the same style as schemes eventually compiled in Walton *et al.* (2008). Sense-making systems aid human users in properly reasoning about an issue. The insights of Bex *et al.* (2003) can be seen as complementary to sense-making systems that organize evidence in other domains such as the clinical decision support (CDS) system implemented by Fox *et al.* (2010) in the domain of medicine. It is our intention that EFA will eventually provide an additional tool to sense-making systems for evidence evaluation, specifically for evidence that can be called *quantifiable* and *verifiable*, as will be defined and discussed later. Another approach to evidence-based reasoning in the domain of law is automated legal reasoning on a given evidential knowledge base. An approach of this kind can be found in Keppens *et al.* (2003), which uses a knowledge base of evidence to automatically generate crime scene scenarios and is a type of automated Decisions Support System (DSS). Another automated evidence-based DSS can be found in Liu *et al.* (2008), in the domain of business.

EFA complements such systems by providing a general framework for evidence-based reasoning which is relevant to legal and business reasoning but also encompasses evidence-based reasoning that could be applied to other domains such as medicine, engineering, science, and politics. This dissertation presents one type of engineering application in the domain of Sensor Webs. The purpose of EFA, as presented in this dissertation, is not to supplant other types of argumentation (or even other approaches to evidence) that can be expressed using the ASPIC framework. It is to provide an argumentation tool specifically for dealing with knowledge bases of quantified sensor data. A knowledge base that is limited to this type of data has special properties that the EFA approach can utilize.

3.1 Core Concepts

The title and basic principles of this approach are inspired by an epistemological position that, in the philosophical literature, is called "Evidentialist Foundationalism", "Foundational Evidentialism", or just "Evidentialism" (Black, 2008; Conee *et al.,* 2004; Poston, 2007; Wampler-Doty, 2010). The central idea of the Evidentialist approach to epistemology is that all of an agent's justified beliefs are based on evidence. Or more formally: Agent S is justified in believing proposition p at time t if and only if S's evidence for p at t supports believing p.

To stress the importance of evidence as the foundation for arguments used in this system, the name Evidentialist Foundationalist Argumentation (EFA) will be used. Also, the incorporation of the term "Foundationalist" highlights the association of this argumentation theory with the epistemological position of Foundationalism, which is summarized in Haack (2009). The exclusive use of *firm* arguments by an epistemology is a type of Foundationalism.

3.1.1 Quantifiability and Verifiability

Two of the central requirements of EFA are that evidence should be *quantifiable* and *verifiable*. They will be semi-formally defined now.

Definition 17: [Quantifiability] Let P be the set of all propositions. $p \in$ P is *quantifiable* iff p can be represented by a series $(b_0,\ldots,b_n)$ where $b_i \in \{0,1\}$.

By this definition, a proposition is *quantifiable* if it can be represented by a series of bits. So a temperature sensor reading, a digital photograph, or a recorded sound are all types of quantifiable propositions. In addition, any string is quantifiable. Also, the results of any calculations in a computer system are quantifiable. Any datum that can be represented by bits in a computer system is quantifiable.

Definition 18: [Verifiability] Let Q = $\{q : q$ is *quantifiable*$\}$. Let a *sensor* be any device that quantifies environmental input. $p \in$ Q is *verifiable* iff p is directly produced by a sensor.

And, by extension, these two foundational definitions can be used to define *evidence*.

Definition 19: [Evidence] Let E = $\{e : e$ is *verifiable*$\}$. $p \in$ P is *evidence* iff $p \in$ E. When interpreted as directly representing only a *sensor reading*, E $\subseteq K_n$.

By Definition 18, a proposition is *verifiable* if it is a quantifiable proposition produced directly by a sensor. This includes data traditionally labeled as sensor data such as video, digital photographs, temperature sensor data, wind sensor data, or air pressure sensor data.

It is important to clarify what this evidence actually directly represents. For example, if a temperature sensor reads 35 degrees celsius at the coordinate (54, 45) at time 12:01, this in *not* direct evidence that any of those values actually reflect reality. Perfect sensors are, for all practical purposes, impossible. The temperature may have actually been 2 degrees colder, the location may have actually been a few meters further in both directions, and the time may have actually been 12:00:56. But this is direct evidence that the sensor *read* those first values. Sensor data are direct evidence of *what sensors read*, as opposed to direct evidence of *what actually happened*. Arguments about *what actually happened* are defeasible arguments based on sensor data. Using this interpretation of evidence, $E \subseteq K_n$. Meaning, evidence, as defined and interpreted here, is treated as a type of axiom, where *axiom* is taken to mean "a self-evident truth that requires no proof ". The reading of a sensor at a certain point in time will never change. It will always be true that the sensor read that reading at some time. That is why sensor readings can be classified as axioms.

In addition to including traditional sensor data, Definition 18 also includes input not traditionally labeled as sensor data, such as input from a mouse and keyboard. By this definition, the string "Unicorns exist", as typed into a text file or website forum, is considered verifiable. This definition may at first appear to counter-intuitively support the policy that simply typing "Unicorns exist" can be considered direct evidence that unicorns exist. It does not. That string is merely direct timestamped evidence that someone *typed* "Unicorns exist" (or wrote a program to produce the string). If desired, this evidence can then be used to support a defeasible EFA *argument* that unicorns exist. But such an argument would not come close to having the *strength* of an EFA argument that could be constructed for an entity such as bears or bacteria. Specific types of EFA arguments and their associated strengths will be defined later.

Definition 18 can also include evidence indirectly related to human cognition and emotion. For example, fMRI scans and recorded spoken reports of emotions can be used as evidence to form a defeasible argument that a person experienced a certain emotion at a certain time. While emotions themselves cannot be used to directly justify arguments in EFA, defeasible arguments that are *about* emotions (such as what should be done in a situation to cause someone to experience happiness) can include indirect evidence for emotions.

An objection can be made to the requirement of quantifiability. This requirement causes EFA to exclude many types of physical items that have traditionally been considered "evidence", such as written documents, polaroid pictures, stone tablets, and ancient artifacts. Indeed, such items must be excluded from EFA because it is a computational system and such objects cannot be directly interpreted by a computational system. However, because all of these types of items can be rapidly represented by quantified data produced by sensors such as digital cameras, flatbed scanners, and even three-dimensional scanners, their exclusion from being directly included as evidence is only a temporary barrier to their representation within EFA.

These restrictions on what can be called evidence bring with them some useful properties. One is that all *conclusions* in EFA are *quantifiable* and can be *calculated* directly from evidence. Another is that all arguments are either firmly based on axioms from K_n or are derived from the conclusions of such arguments.

3.1.2 EFA Argumentation Theory

With the concepts of evidence and quantifiable propositions used by EFA defined it will now be unified with the framework of Prakken (2010) to form an AT, starting with the rules used in EFA.

Definition 20: [Calculation Rule] Let Q be the set of all quantifiable propositions. Given $r \in R_d$, r is a *calculation rule* iff r can be represented by a tuple (Gate, Calc) such that

- Gate is a **gate function** that maps Q to $\{0, 1\}$. Gate(p) returns 1 if p fits the criteria of Gate and 0 otherwise
- Calc is a **conclusion calculation function** that maps 2^Q to Q. Given S $\subseteq$ Q, Calc(S) returns a conclusion q
- Given $p_0, \ldots, p_n$, if Gate(p_i) holds $\forall p_i$ and Calc($\{p_0, \ldots, p_n\}$) = q, then it can be said that $p_0, \ldots, p_n \Rightarrow q$

So premises are only accepted by a calculation rule if they meet the criteria of the Gate function. Once they are accepted, they can be used to calculate the conclusion of the rule using the Calc function. This allows the calculation of an arbitrarily complex conclusion in the set of Q based on premises that have been vetted by an arbitrarily complex Gate function. Through the concept of the calculation rule, EFA formalizes a method to calculate conclusions directly from their premises. With these concepts defined, the argumentation system used in EFA can be specified using the form specified by Definition 1.

Definition 21: [EFA Argumentation System] An *EFA system* is an argumentation system AS = (L, Contr, R, $\leq$) where

- L = Q
- Contr is a contrariness function from Q to 2^Q
- R = R_c where R_c is a set of calculation rules
- $\leq = \emptyset$

So the logical language is Q and Contr maps quantifiable conclusions to other sets of quantifiable conclusions. Contrariness in EFA will be used to define argument growth in EFA later. All rules used by an EFA system must be calculation rules. The partial preorder for defeasible rules is empty because all ordering can be done through argument strengths. With the EFA system defined, evidence-based arguments can now be defined based on the forms of Definitions 6-9.

Definition 22: [Evidence-Based Argument] Let E be the set of all evidence. Let R_c be the set of all calculation rules. An argument A is an *evidence-based argument* iff

- $\mathrm{Prem}(A) \subseteq E$
- $\mathrm{Conc}(A) \in Q$
- $\forall A_i \in \mathrm{Sub}(A)$, A_i is an evidence-based argument
- if A is not a premise-encapsulating argument, then $\mathrm{TopRule}(A) \in R_c$

So evidence-based arguments are either defeasible arguments firmly grounded in evidence, or based on other such arguments. And any rules that are employed by such arguments must be calculation rules. The strength of an evidence-based argument, which is integral to ordering arguments, can now be defined.

Definition 23: [Strength] Let A be an evidence-based argument. The *strength* of A is $|\mathrm{Prem}(A)|$, the cardinality of its supporting premises. The strength of A is given by the function Strength(A).

So the strength of arguments in EFA is equivalent to the size of the set of their supporting premises. At first glance, this may appear to be an overly simplistic definition of argument strength, incapable of handling complex argument strength issues. However, some reflection on how arguments are formed reveals it is not. Through the use of an arbitrarily complex Gate function employed by the calculation rules used to build arguments, an EFA AS has fine-grained control over what evidence can be employed as a premise to an argument. This has the potential to cause arguments to differ dramatically in strength, sometimes by orders of magnitude, all based on what evidence can be used to support them by meeting the criteria of the particular Gate function in question.

Also, it is important to remember that this definition of argument strength is applied specifically in the formal domain of arguments built on *quantifiable* and *verifiable* evidence. The premises of EFA are not qualitative, nuanced opinions that can be subjectively interpreted as having varying levels of strength: they are direct sensor readings. In such a framework, the sheer volume of independent sensor readings fitting a specific criteria create the fabric on which higher-level reasoning is built. This concept of strength can be used to specify an ordering of arguments based on the form of Definition 11.

It can be argued that using the summation of evidence for argument strength is inappropriate since, for example, the summation of evidence in a Sensor Web is sensitive to the distribution of the sensors. For example, assume that there are two different geographical areas, the first area having a higher density of sensors than the second area. For two equivalent blizzards, the first area will have more evidence (and therefore a higher argument strength) than the second area. And, therefore, emergency personnel may unfairly give preference to the first area. This example does not negate the validity of using summation of evidence as the basis of argument strength. The reason is, without a higher density of sensors in the second area, there is no way to *justify* that its blizzard is as strong as the one in the first area. The readings in the second area could also happen for a blizzard that is decisively *weaker* than the one in the first area. Therefore, if emergency personnel want to eliminate the possibility of bias, the solution is not to disregard the practice of using evidence summation as the basis of argument strength: the solution is to add more sensors to the second area so that the two areas have equal sensor density. Because *without* that modification and the subsequent evidence it would provide, there is no way to *distinguish* whether the blizzard in the second area is the same strength, stronger, or weaker. If the emergency personnel refuse to do this or cannot afford to, then they must accept the possibility of responding to a weak blizzard because they have to test for a strong one. There is no getting around the fact that, without evidence provided by a higher density Sensor Web or some other sensing source, there simply is no way to know.

Definition 24: [Evidence Amount Ordering] Let $\mathscr{A}$ be a set of evidence-based arguments. Then a partial preorder $\preceq$ on $\mathscr{A}$ is an *evidence amount ordering* iff

(1) $A \prec B$ iff $\text{Strength}(A) < \text{Strength}(B)$

(2) $A \preceq B$ iff $\text{Strength}(A) \leq \text{Strength}(B)$

This definition clearly fits the admissible argument ordering criteria in Definition 11, since those restrictions are only applied to strict rules and no strict rules are used in EFA. With an argument ordering specified, EFA theories can no be defined using the form specified by Definition 12.

Definition 25: [EFA Argumentation Theory] An *EFA theory* is an argumentation theory $AT = (AS, KB, \preceq)$ where AS is an EFA system, $KB \subseteq E$ is a knowledge base in AS, and $\preceq$ is an evidence amount ordering.

Given this definition for an EFA theory, Definition 16 as reproduced from Prakken (2010) holds without modification, demonstrating the utility of Prakken's definitions. Namely, given arguments A and B from an EFA theory, A defeats B if Conc(A) is a contrary of any of the conclusions of B's sub-arguments and $B \prec A$ ($\text{Strength}(B) < \text{Strength}(A)$). If $\text{Strength}(A) = \text{Strength}(B)$ and both arguments attack each other, then by Prakken's definitions neither argument defeats and the situation can be informally considered a tie.

An important purpose for EFA is to aid in investigations undertaken by agents. Evidence-based arguments are used as a basis for building and storing knowledge. When arguments come into conflict, such as when two different arguments propose two different opposing actions for an agent, then the conflict can be resolved using evidence amount ordering. The evidence amount ordering is likely to be useful because arguments cannot gain strength without gaining premises. And an agent cannot gain premises for its arguments without discovering evidence that meets the criteria of the Gate function used in the calculation rules of the evidence-based arguments.

This highlights the importance of building an effective Gate function. For example, in many agent system applications, it would be undesirable to have very similar evidence count twice for the same argument. For example, if an agent is advancing the argument that a wide field of land has a high temperature, it would likely be undesirable to allow an agent to read multiple temperature readings in one location within a matter of a few seconds and count each individual reading as a new piece of supporting evidence for its argument. A good Gate function would accept only one such reading and reject others as too similar to the first reading.

With the formal foundations of generalized EFA established, the specific application used in this work can be specified.

3.2 Application to Sensor Webs

The general definitions from the previous section can be applied to any area to which sensor evidence from the physical world is relevant. As detailed before, this includes the domains of medicine, engineering, science, and politics. In this section, the general definitions will be applied to the Sensor Web problem domain, as first described in Redford and Agah (2010). This requires the instantiation of concrete data structures that meet the requirement of the formal definitions. It also requires data structures for creating new arguments as new evidence is gathered by the agents in the system.

In the application of EFA to Sensor Webs, all arguments represent geophysical events which cover a specific geographical area and have certain physical properties. Here the type of evidence used by the system is established. The evidence-based arguments and sub-arguments used by the system are also established.

3.2.1 Observations

As specified in Definition 22, all premises for evidence-based arguments must be evidence as defined in Definition 19. The evidence used in this application is in the form of *observations* in the form of a tuple (W, V, L), where W is a weather property, V is a floating point scalar value, and L is a location. As required by Definition 19, these observations come from sensors. For the experiments in this paper, W is either wind chill, wind speed, or visibility. V is the value associated with that property, degrees celsius for wind chill, kilometers per hour for wind speed, and meters of viewable distance for visibility. L is a 2-dimensional global coordinate. For example, "(wind chill, 40.1, (54, 49)" and "(wind speed, 35, (35, 40))" are valid observations. Examples of the observation types, as they will be referenced visually, are shown in Figure 11.

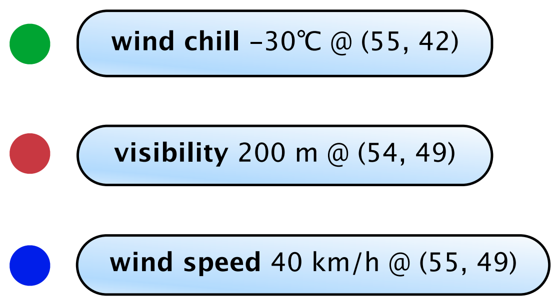

Figure 11. Example observations, the form of evidence in this application.

3.2.2 Fields

A *field* is an evidence-based argument whose calculation rule's conclusion is a tuple (R, C, S), where R is a circular region, C is a weather condition, and S is a strength value. R is a tuple (L, r) where L is a 2-dimensional global coordinate and r is a radius in kilometers. C is a tuple (W, Op, V) where W is a weather property as explained in the previous section, Op is a comparison operator, and V is a value as explained in the previous section. S is the strength value of the argument. For example, "([(55, 56), 3.2], [wind speed, ≥, 35.0], 44)" is a valid field conclusion, stating that a field for wind speed greater than 35 km/h exists at (55, 56) covering a radius of 3.2 with a strength of 44 matching observations. An example of a field, as it could be represented visually, is shown in Figure 12.

Figure 12. Visual example of a Low Wind Chill Field with strength 4.

The field's calculation rule's Gate function, matching the form specified in Definition 20, requires observations to (1) meet the observation field's weather condition, (2) be physically close enough to previously accepted premise observations, and (3) be distinct enough from previously accepted premise observations. The calculation rule's Calc function uses the premises of the field to geometrically calculate the region and cumulatively calculate the strength for the field's conclusion.

A simple field argument is shown in Figure 13. In the figure, multiple wind chill observations meeting the criteria of the field's Gate function are used to calculate the conclusion for a Low Wind Chill Field using the Calc function. In practice, fields can actually contain hundreds of observation premises spread across a wide geographical area. For example, on one of the largest maps used for preliminary testing of EFA, a high wind speed field in the middle of a sizable blizzard associated with the weather condition “(wind speed, >, 40)” produced fields with a strength of 523 spread across a region with a 20 km radius. This example argument conceptually matches with the visual representation shown in Figure 12.

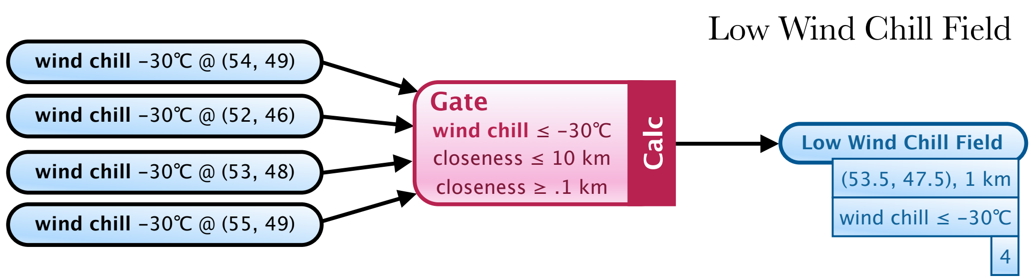

Figure 13. Example field argument.

3.2.3 Phenomena

A *phenomenon* is an evidence-based argument whose calculation rule’s conclusion is a tuple (R, C, S), with the same type of values as used for field arguments, except that C is a set of weather conditions rather than a single weather condition. Another difference between a field and a phenomenon is that a phenomenon has one or more fields as sub-arguments, each matching one of the weather conditions in C. Phenomena represent high-level weather occurrences like blizzards, hurricanes, or thunderstorms. The only type of phenomena used for experiments in this dissertation are blizzards. A blizzard has a value of C corresponding to the conditions “wind chill ≤ -30 degrees celsius”, “wind speed ≥ 40 kilometers per hour”, and “visibility ≤ 400 meters”, which are the conditions for a blizzard as specified by Environment Canada (2011). An example of a phenomenon, as it could be represented visually, is shown in Figure Figure 14. In the figure, the model contains low wind chill fields in green, high wind speed fields in blue, and low visibility fields in red. A blizzard phenomenon is labeled with a dashed brown line, supported by underlying fields of all three types. All fields and phenomena are associated with the numerical strength of evidence supporting them.

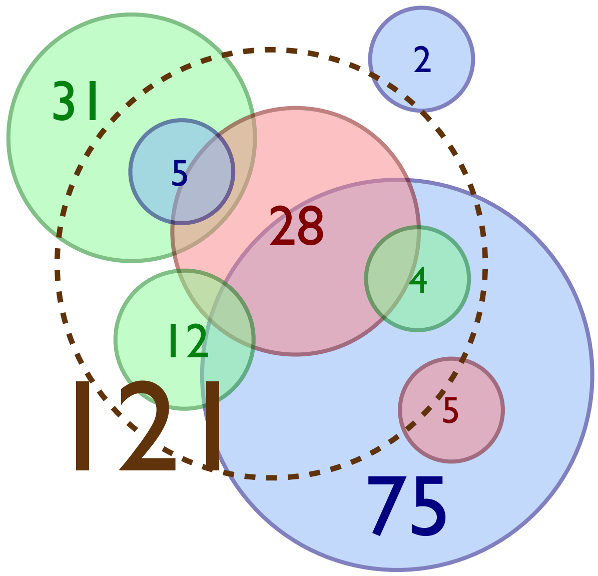

Figure 14. Visual example of a Blizzard Phenomenon with strength 121, along with its supporting fields.

A phenomenon's calculation rule's Gate function requires the conclusions calculated from its field sub-arguments' premises to (1) meet one of the phenomenon's weather conditions, (2) be physically close enough to the regions of previously accepted field arguments, and (3) be distant enough from previously accepted field arguments with the same weather conditions. The calculation rule's Calc function uses the conclusions calculated from the premises of the accepted field arguments to geometrically calculate the region and cumulatively calculate the strength for the phenomenon's conclusion.

A simple phenomenon argument is shown in Figure 15. Note that only the field conclusions of the fields in Figure 15 are shown to allow for space. In the actual system, the field arguments, as shown in Figure 13, would also be part of the phenomenon argument as its sub-arguments. In practice, phenomena can actually contain multiple field conclusion premises of the same type. For example, two large high wind speed field conclusions can cover a large area containing 10 different low visibility field conclusions, making them all part of the same blizzard argument since the high wind speed fields overlap with the various low visibility fields.

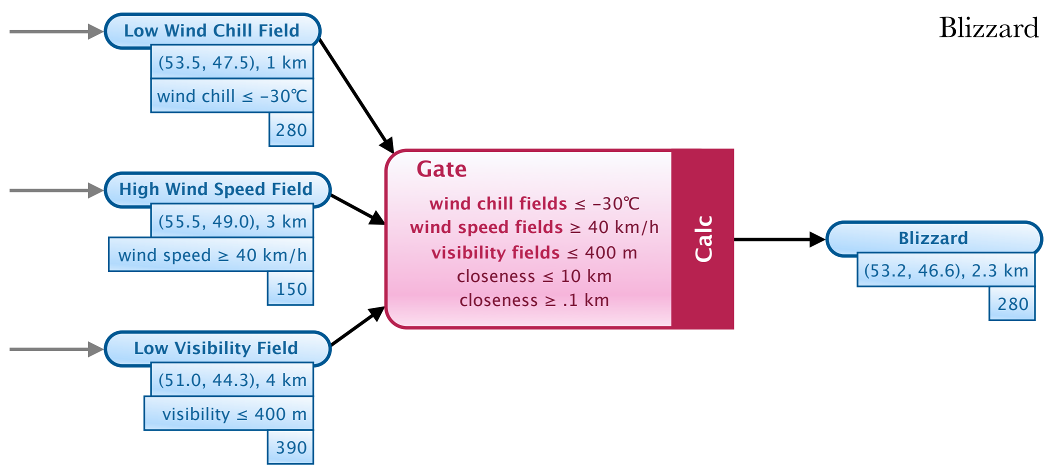

Figure 15. Example phenomenon argument.

It can be argued that the system could benefit from the use of fuzzy logic, which has been employed successfully in many practical engineering applications for decades (Fox, 1981; Madau *et al.,* 1993; So *et al.,* 1993). For example, given a weather condition of "wind speed ≥ 40 km/h" and an observation with a wind speed of 39.9999, the observation would be rejected by the current version of the system. A fuzzy logic version of this system could include that observation under fuzzy membership.

There are two reasons why fuzzy logic is not used in this dissertation. The primary reason is that the official definition of “blizzard” given by Environment Canada (2011) is not defined in fuzzy terms. Since the definitions given by meteorological organizations have strict cutoff points, so does this system.

The other reason is that, for this application, we think that the information a fuzzy system would communicate can still be captured by this system. This is because all categorization systems, even fuzzy systems, must have some implicit cutoff point at which they stop accepting membership. For example, we can assume that the system is redefined to use “wind speed ≥ 50 km/h” and allow fuzzy membership from 40 km/h to 50 km/h. In this case, a reading of 39.9999 wind speed would *still* be rejected, as shown in Figure 16. Even fuzzy systems have implicit boundaries.

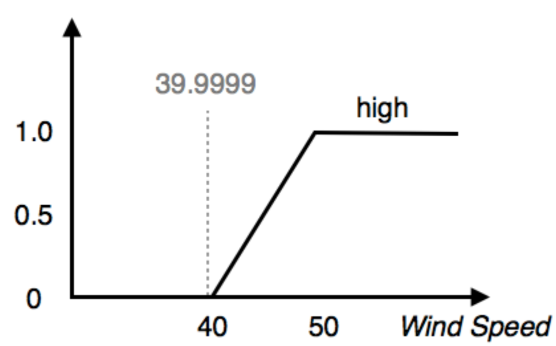

Figure 16. Even fuzzy systems have implicit boundaries.

While fuzzy systems obviously have many useful engineering applications, they do not appear to be necessary for this application. Using the techniques already provided by this system, the type of information that a fuzzy system would provide (e.g. “low wind speed”, “high wind speed”, and a linear gradation between) can still be fully distinguished simply by categorizing the relevant property in the same way a fuzzy system would. For example, if the average wind speed of a blizzard is 50 km/h or higher it can be said to have “high wind speed”. If it is only 40 km/h, it can be said to have “low wind speed”. If it is between 40 km/h and 50 km/h, it can be treated accordingly linearly on the scale between the two. This can all be captured by a linear function in an Inference node or an Argument’s conclusion calculation function. An explicitly fuzzy system is not necessary for this distinction if making it is somehow useful to the agents’ conclusions. For this application, the distinction between “low” and “high” wind speeds among accepted observations does not seem useful or necessary.

Rather, fuzzy systems seem to excel in applications where systems need to gracefully switch between two different modes because human users require gradual change for safety or aesthetic reasons, such as in anti-lock brakes or camera focusing (Madau *et al.,* 1993; So *et al.,* 1993). Such time sensitive gradation to meet human sensibilities is not necessary for this application, which is primarily meant to calculate and categorize relatively slow-changing weather properties over time (on a scale of minutes rather than seconds or milliseconds).

3.2.4 Collections

In terms of the implementation of EFA theory, enough details have been specified. However another type of notable data structure is required for the efficient implementation of this system and deserves mention.

As agents in using this EFA theory gather observations, they feed those observations to *collections* which dynamically re-evaluate what fields and phenomena can be formed from the collected observations. For example, disparate wind speed observations may start far enough apart that they can be used to create their own separate field arguments. However, as an agent continues to feed new wind speed observations to the field collection, these observations may be merged into a single field argument, due to meeting the closeness constraints of its Gate function when combined with the new wind speed observations. Such a scenario is shown in Figure 17. Collections allow constant re-evaluation of the arguments that can be formed by an agent as it gathers new observations.

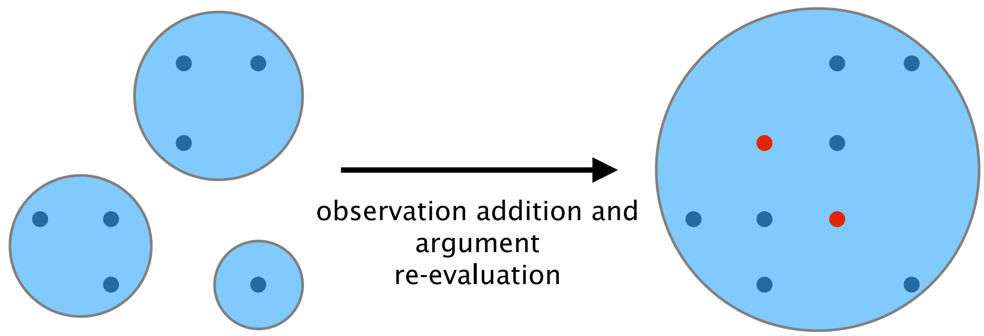

Figure 17. As new observations are added to a collection, the collection re-evaluates the possible arguments that can be formed.

4 Experiments

In this chapter, we will describe the materials used in our experimental setup. Specifically, we will explain where our real-world data comes from and what tools were used to develop the Multi-Agent Systems tested. We will also present the technical details of the metrics used.

4.1 Technology

Our goal for the experiments was to test the ability of a Multi-Agent System to accurately characterize the properties of blizzard conditions. The first step was to determine the quantitative evidential conditions for a blizzard. From the definition of blizzard by Environment Canada (2011), we used the conditions of *wind speeds* 40 km/h or higher and *visibility* 400 m or less for at least 4 hours. Because we wanted to track the worst blizzards as well as challenge the reasoning capabilities of our sensor agents, we added the additional condition of *wind chill* dropping to -30 degrees celsius or lower.

Because our work concerns in-situ Sensor Webs, which cover a broad geographical area, in addition to scalar values for sensor readings, we also had to consider the spatial properties of blizzards. We defined a **blizzard** spatially as the union of overlapping geographical areas with the scalar properties mentioned above.

To rigorously test the validity of our systems, we also define the concept of a **near-blizzard** as a weather scenario where some but not all of the above scalar conditions occur or all of them occur but do not geographically overlap. Finally, a **non-blizzard** is defined as a weather scenario where only one or none of the above scalar conditions occur.

Using the weather functions of Wolfram Mathematica (2011), we retrieved the historical wind speed, visibility, and wind chill data for 30 blizzards, 30 near-blizzards, and 30 non-blizzards from various locations including Russia, China, and the United States that occurred from the years 2000-2010. The data are based on the historical data taken from geographically distributed weather stations in these countries. An example of retrieved Mathematica data and how it could be modeled by an agent who has received some of its observations is shown in Figure 18.

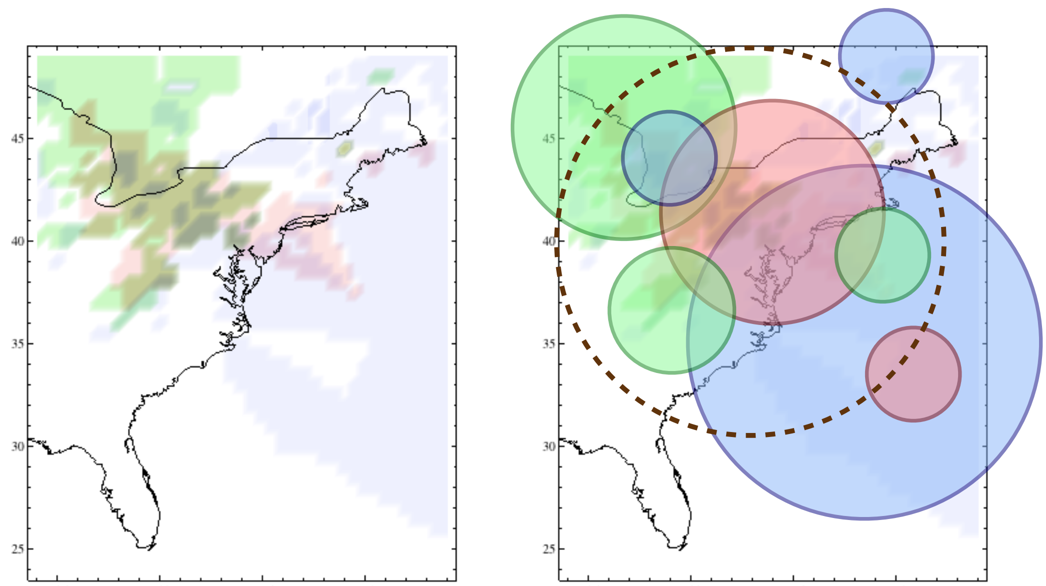

Figure 18. Example of historical weather data and how it could be modeled by an agent who has received some of its observations.

We also wanted to test the scalability of the agent systems so we tested each scenario for agent population sizes of 25, 36, and 49. Perfect squares were chosen to maintain a square shape to the grid of in-situ sensors associated with the agents. As mentioned earlier, each sensor station is associated with a single agent. These grid sizes were chosen because they represent perfect square grids of increasing size. This results in a total of 270 test scenarios.

The EFA evidence-based argument construction and reasoning used by our agent systems were developed in Java and can be used independently from the agent systems. Due to the hierarchical and extendible nature of the classes implemented, the system relies heavily on inheritance and generics.

The agents systems used for the experiments were built using the Jadex BDI Agent System (Jadex, 2011). Jadex is built on top of the Jade Java Agent Development Framework (Jade, 2011).

Jade provides an Agent Management System (AMS) on which agents run. Agents find each other using a Directory Facilitator (DF) . In the tested systems, the DF associates agent coordinates with their agent ID. Jade implements FIPA-ACL compliant messages, which are used by both tested systems. The use of FIPA-ACL by the implemented agents to register with and search the Jade DF is shown in Figure 19.

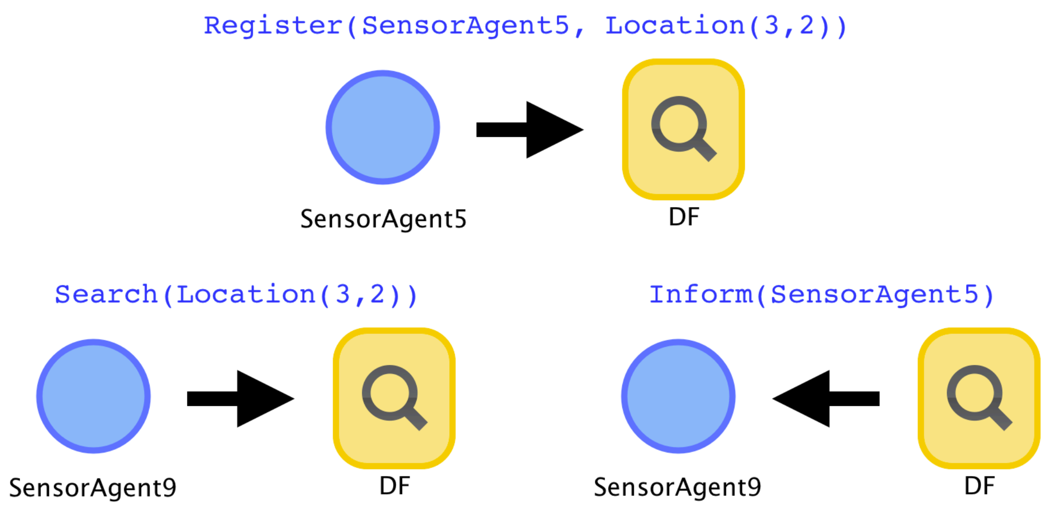

Figure 19. Example of FIPA-ACL messages used by the implemented agent systems.

Jade has been developed to run on mobile devices with batteries, low processing power, and wireless communication such as basic cell phones, so implementing it on in-situ sensors that communicate wirelessly in real time is possible.

4.2 Metrics

Our experiments evaluate and analyze the performance of a group of agents, each of which is associated with sensors in a grid, forming an *in-situ* sensor web. Each sensor in the grid has access to wind chill, visibility, and wind speed data. As described in the previous section, data are is provided by Wolfram Mathematica (2011). The goal of the agents is to achieve the highest possible global accuracy, i.e., the accuracy of each individual agent's model of the geophysical situation. Since each agent only has access to the observations of its individual sensors, it must communicate with the other agents to have an accurate model. We have established a number of metrics by which to measure the performance of the agents in this experimental setup.

Whenever the agents communicate, there is a message cost associated with their communication. This **message cost** is the length of the string required to represent the message, multiplied by the Euclidean distance between the two communicating agents. This definition is based directly on the underlying real world technology that would be used to implement the agents on actual sensors using radio communication. The *length* of the string represents how many bytes would be needed to transmit the message. The bytes break down into bits. Each bit transmitted translates directly into battery power the sensor would consume while transmitting the bit. The *Euclidean distance* represents the radius of the broadcast a sensor would have to extend with its radio communication to reach the other agent. The greater the broadcast radius, the more power the sensor will consume. This distance cost will apply to each bit of data transmitted, which is why the Euclidean distance is multiplied by the length of the string. The conceptual basis for message cost is shown in Figure 20. The **message amount** metric is just the total number of messages sent.

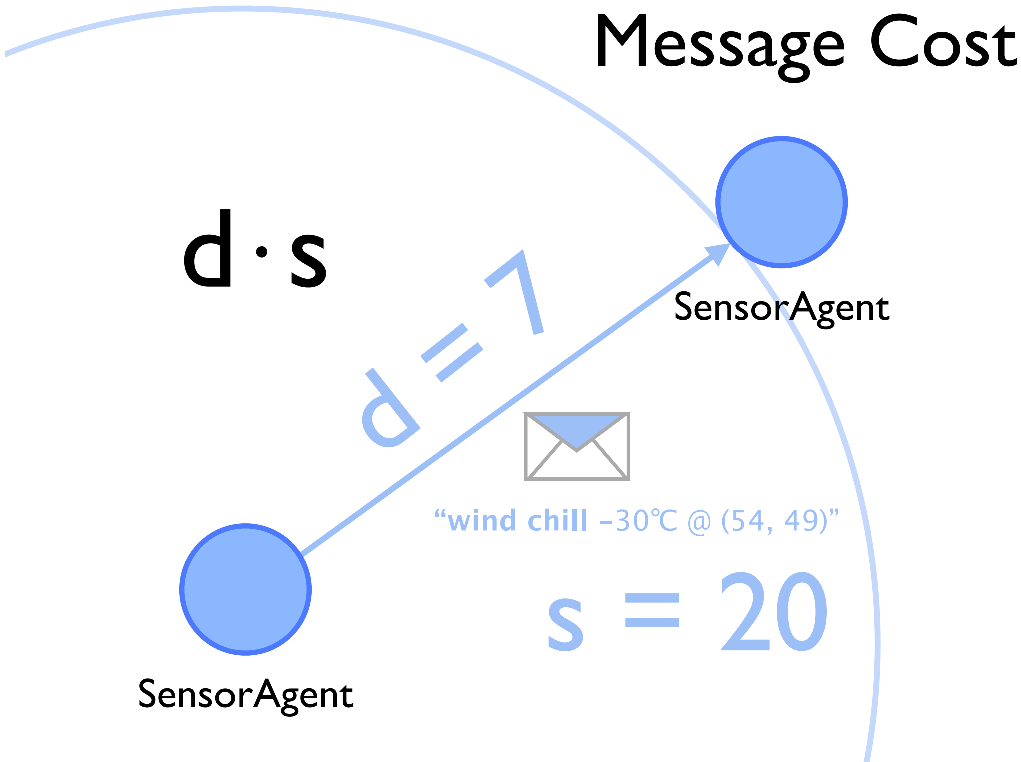

Figure 20. Message cost is calculated as the distance of communication times the length of the string communicated.

The **total accuracy** of each individual agent's model, represented by a set of (R, C, S) tuples, is determined by its similarity to the evaluator agent's model. The *similarity* of an agent's model to the evaluator agent's model is an average of the similarity between the regions and strengths of each model. The similarity of two regions is the area of their intersection divided by the area of their union. The similarity of two strengths is the smaller strength divided by the larger strength. Only the regions and strengths of fields using the same weather condition are compared. Whenever similarity between two sets of fields and phenomena are calculated, the most similar fields and phenomena are compared and eliminated first. If the agent's model disagrees with the evaluator agent's model on the number of fields or phenomena, a penalty of zero is applied to the average. For example, if the agent thinks there are four fields and the evaluator thinks there are five, the agent can achieve at most 80% similarity. All similarity values are percentages. The total accuracy metric is illustrated in Figure 21.

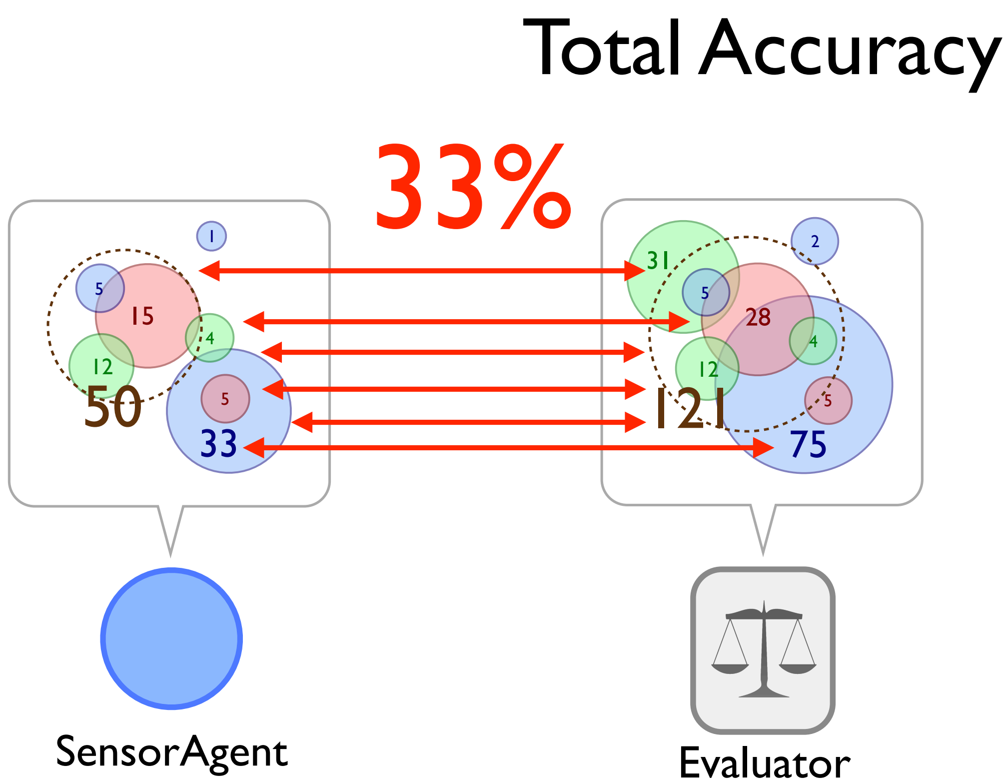

Figure 21. The total accuracy metric requires correspondence between each individual component in an agent's model with the evaluator's model, making it the most detailed accuracy metric.

The *super region* of an agent's model is calculated by combining all of its field and phenomenon sub-regions into the largest encompassing region. For the CNO agent system, the R′ produced by the agent's Kalman Filter is simply evaluated directly. The **super region accuracy** of each individual agent's model is the similarity of its super region to the super region of the evaluator agent's model. The super region accuracy metric is illustrated in Figure 22.

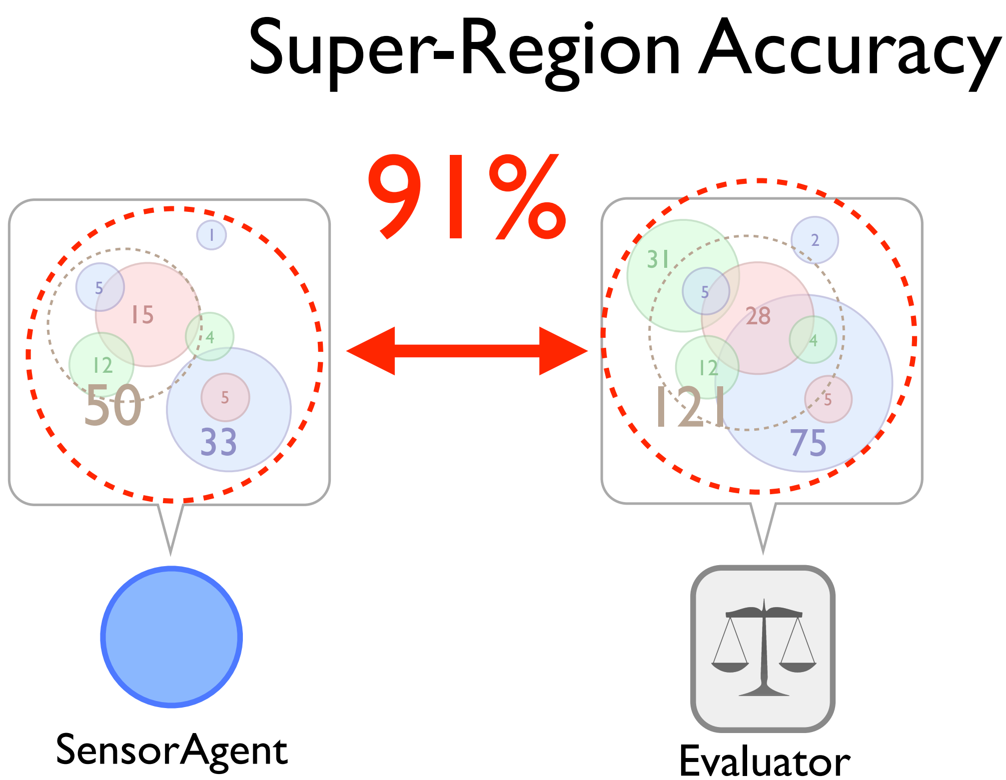

Figure 22. The super region accuracy metric requires correspondence between an agent's calculation of the largest encompassing reason of its model's components with the evaluator's.

The *super strength* of an agent's model is produced by summing all of the strengths of its fields and phenomena. For the CNO agent system, the S′ produced by the agent's Kalman Filter is simply evaluated directly. The **super strength accuracy** of each individual agent's model is the similarity of its super strength to the super strength of the evaluator agent's model. The super strength accuracy metric is illustrated in Figure 23.

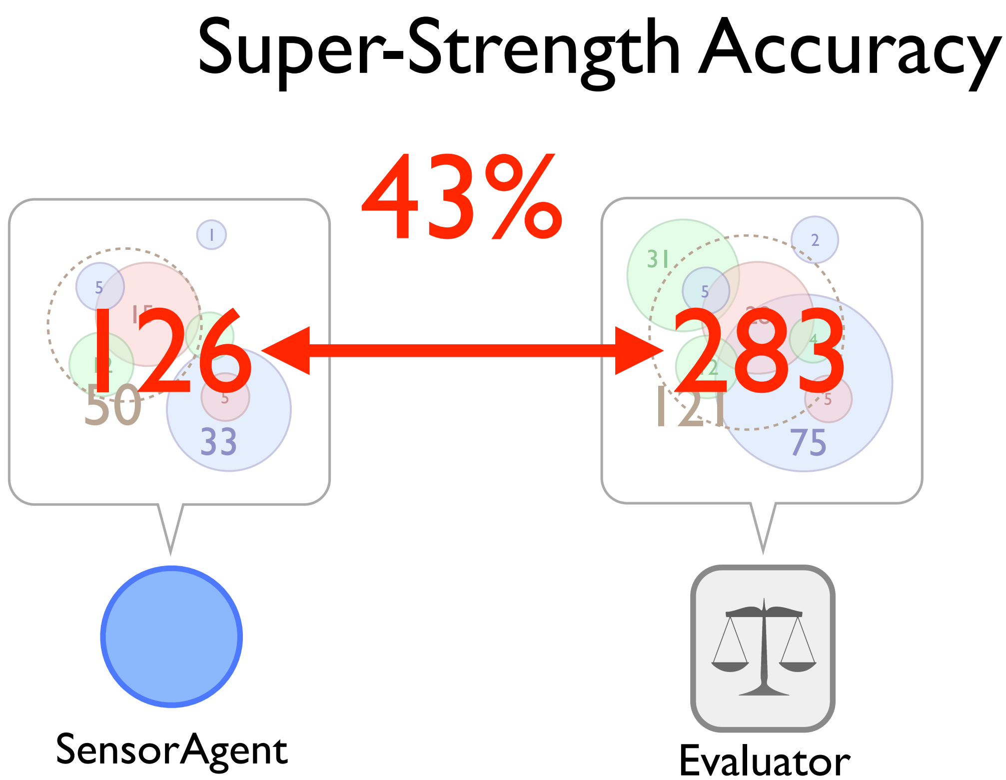

Figure 23. The super strength accuracy metric requires correspondence between an agent's calculation of the total strength of its model's components with the evaluator's.

All of the message-related metrics are summed over all agents in the multi-agent system. All of the accuracy metrics are averaged over all agents. This provides a way of measuring the performance of the entire agent system.

The only types of observations, and therefore the only type of fields, used in these experiments are wind speed, wind chill, and visibility. The only type of phenomena tracked are blizzards.

5 Investigative Argumentation-based Negotiating Agents

In this chapter, we will detail the research approach of **Investigative Argumentation-based Negotiating Agents** (IANA). This system is utilized in the problem domain of Sensor Webs. In the proposed problem, *in-situ* sensors are spread out in a grid across a wide geographical area. Each sensor outpost in the grid is assumed to have three sensors: wind chill, wind speed, and visibility. Each sensor outpost is associated with a single agent. Individual agents in the grid can communicate with other agents through wireless communication. In this way, the agents form a Sensor Web.

The goal of individual agents is to have as accurate a model of the current weather situation as possible, while minimizing communication costs to save battery power. Given a single point of time, the goal of the agent system is to maximize the accuracy of each individual agent's model by strategically sharing observations between agents. This goal represents a type of Decentralized Sensor Fusion (Rosencrantz *et al.*, 2003).

In IANA, the Directory Facilitator (DF) associates agent coordinates with their agent ID. Jade implements FIPA-ACL compliant messages, which are used in IANA. The use of FIPA-ACL by the implemented agents to register with and search the Jade DF is shown in Figure 19.

In addition to the DF, there are two other types of agents used in the experiments: the sensor agents and the evaluator. A **sensor agent** is allocated to each sensor outpost location on the grid. The **evaluator** is a special type of agent that is given all of the weather data from Wolfram Mathematica for the current scenario. The evaluator uses this data to produce the correct arguments and resulting conclusions that can be formed from that evidence. When all of the sensor agents have finished communicating, they send their final argument conclusions to the evaluator agent. The evaluator agent judges the overall performance of the agent system based on criteria that will be specified in the Metric section of the Materials chapter.

The sensor agents in IANA attempt to solve the task of maximizing accuracy while minimizing message cost in two primary stages. In the first stage, a sensor agent reads its three local observations (wind speed, wind chill, visibility) and adds them to a collection. The agent then checks the collection to see if any of these observations resulted in the formation of a new field argument. If none of them resulted in a new field arguments, the agent does not contact any other agents and simply waits to be contacted by other agents who *were* able to create field arguments. On the other hand, if any observations *did* result in the formation of a new field argument, the agent enters into the second stage of the algorithm. The practical idea behind this first stage is that an agent should only attempt to communicate observations that are *relevant* to building an argument.

In the second stage, the agent generates a *situation summary* from its collection, which is the tuple (P, F, S, R). P is the number of phenomena in the collection. F is itself a tuple (Wc, Ws, V) where Wc is the number of low wind chill fields, Ws is the number of high wind speed fields, and V is the number of low visibility fields. S is the combined strength of all arguments in the collection, representing the total evidence of the collection, called the *super strength*. R is the minimum circular region that can encompass all of the observations in the collection, called the *super region*.

The agent then enters into what will be called the Share on Disagreement (SoD) protocol with each of the other agents in the system. In the SoD protocol, the Initiator agent sequentially shares each individual item in the situation summary with the Participant agent. The situation summary tuple has been strategically ordered to place the most important and most contentious (most likely to cause disagreement) items first. This is meant to minimize time spent communicating. If, at any point in the protocol, the Participant disagrees on an item in the situation summary, the Initiator sends its relevant observations. Whether a disagreement has occurred depends on the item in question: for P and F, the items must match exactly; for S and R, 75% similarity is acceptable grounds for agreement due to its high performance in preliminary testing. This protocol is shown in Figure 24.

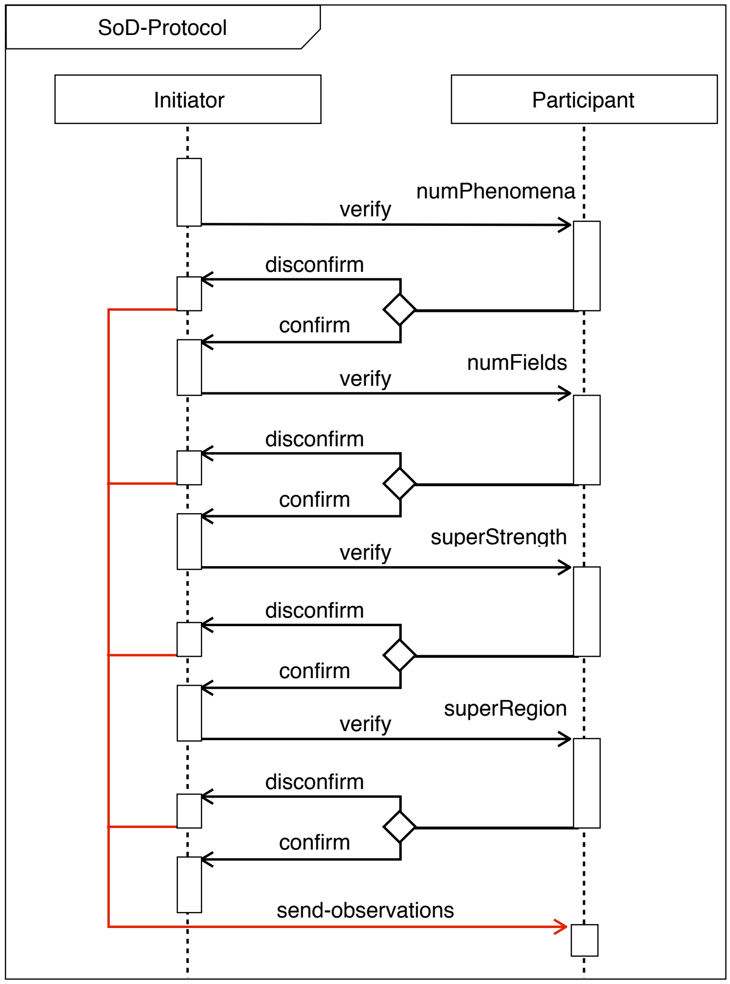

Figure 24. The SoD protocol used by IANA.

The inspiration for the SoD protocol (which depends on the use of EFA) comes from the practical maxim: if agents sufficiently agree on their general conclusions, they do not need to share the premises that caused them to reached those conclusions. As referenced in the Introduction and Figure 1, if two human experts (for example, doctors) agree on their conclusion as to which detailed action is best for a situation, it is not necessary that they share the studies that lead them each individually to that conclusion. It is not unless their is a disagreement that experts should share their evidence and come to a consensus about whose conclusion has the most evidence. In the same way, if agents in a Sensor Web basically agree on their conclusion as to what is happening in the current geophysical situation, then it is not necessary that they share the observations that led them to their conclusions.

This is one of the strengths of argumentation: the ability to *summarize* the results of an argument using its conclusion while *still having access* to the premises used to produce that summary, if necessary. Exchanging the content of conclusions through situation summaries saves message costs if the agents agree; but having access to the premises leading to those conclusions saves accuracy if the agents disagree. Examples of agents in both disagreement and agreement situations are shown in Figures 25 and 26, respectively.

Disagreement

<table>
<tr><th>Initiator</th><th>Participant</th></tr>
<tr><td>agent10 | Initiating SoD with agent14</td><td></td></tr>
<tr><td></td><td>agent14 | Participating in SoD with agent10</td></tr>
<tr><td>agent10 | Confirming: 1</td><td></td></tr>
<tr><td></td><td>agent14 | Received: 1
agent14 | Confirming</td></tr>
<tr><td>agent10 | Confirming: 2</td><td></td></tr>
<tr><td></td><td>agent14 | Received: 2
agent14 | Confirming</td></tr>
<tr><td>agent10 | Confirming: 1</td><td></td></tr>
<tr><td></td><td>agent14 | Received: 1
agent14 | [Disconfirming]</td></tr>
<tr><td>agent10 | [Disconfirmed]
agent10 | Sending Observations:
[{Visibility : 5.63 @ (2, 2)},
{WindChill : -10.58 @ (2, 2)}]</td><td></td></tr>
<tr><td></td><td>agent14 | Received Observations:
[{Visibility : 5.63 @ (2, 2)},
{WindChill : -10.58 @ (2, 2)}]</td></tr>
</table>

Figure 25. Example of SoD disagreement situation in actual IANA execution.

Agreement

<table>
<tr><th>Initiator</th><th>Participant</th></tr>
<tr><td>agent13 | Initiating SoD with agent12</td><td></td></tr>
<tr><td></td><td>agent12 | Participating in SoD with agent13</td></tr>
<tr><td>agent13 | Confirming: 1</td><td></td></tr>
<tr><td></td><td>agent12 | Received: 1
agent12 | Confirming</td></tr>
<tr><td>agent13 | Confirming: 2</td><td></td></tr>
<tr><td></td><td>agent12 | Received: 2
agent12 | Confirming</td></tr>
<tr><td>agent13 | Confirming: 1</td><td></td></tr>
<tr><td></td><td>agent12 | Received: 1
agent12 | Confirming</td></tr>
<tr><td>agent13 | Confirming: 1</td><td></td></tr>
<tr><td></td><td>agent12 | Received: 1
agent12 | Confirming</td></tr>
<tr><td>agent13 | Confirming: 8</td><td></td></tr>
<tr><td></td><td>agent12 | Received: 8
agent12 | Confirming</td></tr>
<tr><td>agent13 | Confirming: {2.3r @ (1, 1)}</td><td></td></tr>
<tr><td></td><td>agent12 | Received: {2.3r @ (1, 1)}
agent12 | [Confirming Final]</td></tr>
<tr><td>agent13 | [Full Confirmation]</td><td></td></tr>
</table>

Figure 26. Example of SoD agreement situation in actual IANA execution.

After all the sensor agents have completed the SoD protocol, they send their final set of (R, C, S) tuples from the conclusions of the field and phenomenon arguments produced by their collections to the evaluator. The evaluator then evaluates the performance of the agent system based on the overall performance of all agents in the system.

6 Complete Data Sharing

The **Complete Data Sharing** (CDS) system shares all sensor observations between all agents. It is our hypothesis that CDS will perform better than IANA for all **accuracy metrics** as well as a **message amount** metric. However, we also predict that these advantages will all have a *small* effect size. It is also our hypothesis that IANA will perform better than CDS for the **message cost** metric and that this advantage will have a *large* effect size.

If these hypotheses are true, then one can argue that, unless maximal accuracy and minimal message amount are required for a system, it is more suitable to use IANA than CDS, due to the strong advantages given by the lower message cost.

In this chapter, we will describe the overall design of the CDS system. Then we will present the experimental results for all metrics, and present the data and graphs for the means, standard deviations, p-values, and effect sizes for each system and agent amount. These descriptions and experiments first appeared in Redford and Agah (2011).

6.1 Design

The primary strategy of the CDS is straightforward: all sensor agents in the system share their three observations (wind speed, wind chill, visibility) at their sensor's location with all other agents.

The interactions of the sensor agents with the DF are the same as those used in IANA and shown in Figure 19. The interactions the sensor agents have with each other and the evaluator are shown in Figure 27. Each sensor agent sends its observations to all the other sensor agents. Once all sensor agents have sent their observations, each sensor agent sends its resulting conclusions to the evaluator.

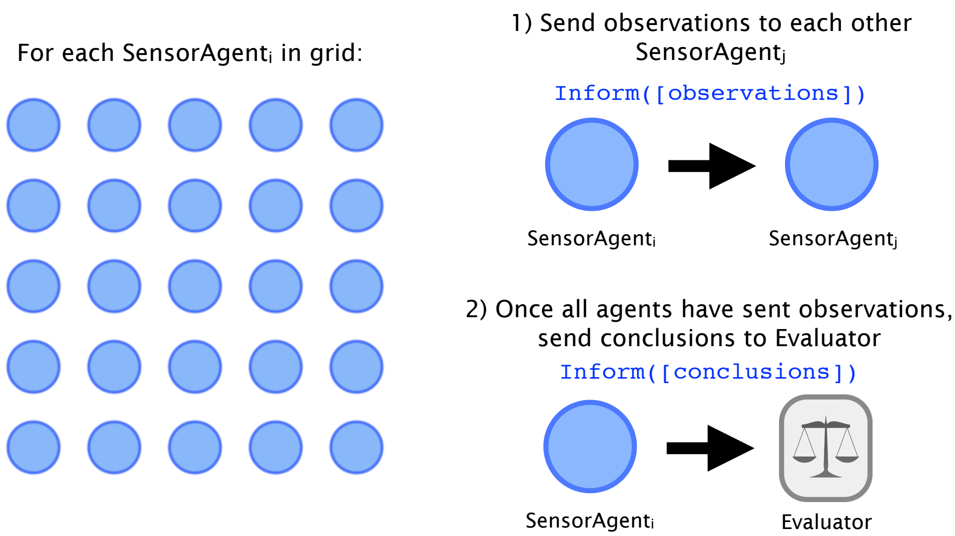

Figure 27. The interactions between agents in CDS.

The advantage of this approach is that it guarantees that all agents will end the experiment with models 100% similar to the evaluator agent's model and will thus produce a score of 100% accuracy for the agent system as a whole for all accuracy metrics. Another advantage of this system in comparison to IANA is that it will have a lower message amount. Agents in IANA use a protocol that requires the exchange of multiple messages between the same agents, whereas agents in CDS send only a single message to each other agent containing their observations. The disadvantage of this approach is that it also guarantees that the agent system will have a message cost that is $O(N^2)$, where N is the number of agents. The purpose of these experiments is to determine the extent of these advantages and disadvantages.

The individual agents in CDS will not have access to any kind of evidence-based reasoning during an experiment. They simply collect and communicate their observations to each other. It is only after an experiment is completed that they feed their observations to a phenomenon collection and send the conclusions to the evaluator agent for evaluation.

6.2 Comparison

Agents in the CDS system always send the same observations for the same amount of message costs for every experimental execution on any geophysical situation. For this reason, we only needed to execute a CDS system for a particular agent amount on each of the 90 geophysical situations once (30 each of different Blizzard, Near-Blizzard, and Non-Blizzard conditions).

CDS has the same metric results no matter if it is working on blizzards, near-blizzards, or non-blizzards. Meaning: it will always have the same accuracy, message cost, and message amount for any given set of situation data. The reason is because all of the agents will always share all of their observations with all the other agents, giving only one possible resulting database and communication pattern for all of the agents in the system. Because of this, only one set of CDS data is represented per metric.

Agents in the IANA system vary in the amount of messages they send to each other depending on the order in which they communicate. Sometimes groups of agents reach a consensus sooner because they communicate in an order that causes mutual consensus. Sometimes this early consensus is premature and causes low accuracy. Sometimes this early consensus is fortuitous and causes high accuracy while maintaining low message costs. Because of the variations in time before consensus for agents, we execute an IANA system for a particulate agent population size ten times for each of the 90 geophysical situations. As discussed in Section 6, this choice produced sufficiently high statistical significance given the random variations in the measured metrics.

However, IANA performs differently depending on whether it is working on a blizzard, near-blizzard, and non-blizzard. For example, non-blizzard geophysical situations require less communication for IANA because fewer arguments can be built because less data is applicable to building any field that can be associated with a blizzard. For this reason, different IANA data are presented for different geophysical situations.

6.2.1 Message Cost

Figure 28 shows the means and standard deviations over all the experiments for message cost. As predicted, the CDS message costs were significantly higher than those for IANA; and the IANA message costs reduced as the activity level of the geophysical situation decreased (e.g., situations with blizzards required more communication than situations with non-blizzards).

Figure 29 illustrates the effect size of the lower message cost advantage that IANA has over CDS, as produced by ANOVA. As predicted by our hypothesis, all of these effect sizes are large (above 0.8) with the exception of IANA's performance for blizzards with 25 agents which was still 0.78, a near-large effect size. The ANOVA established these effect sizes with a p-value less than 0.001, giving them greater than 99.9% confidence.

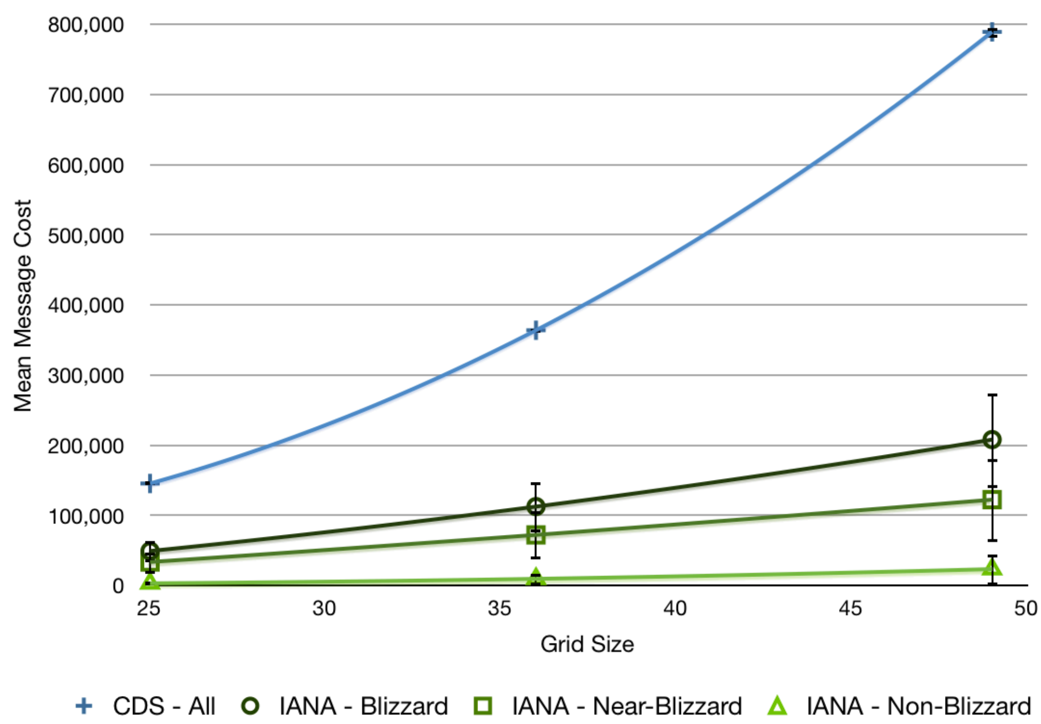

Figure 28. Message Cost Means and Standard Deviations.

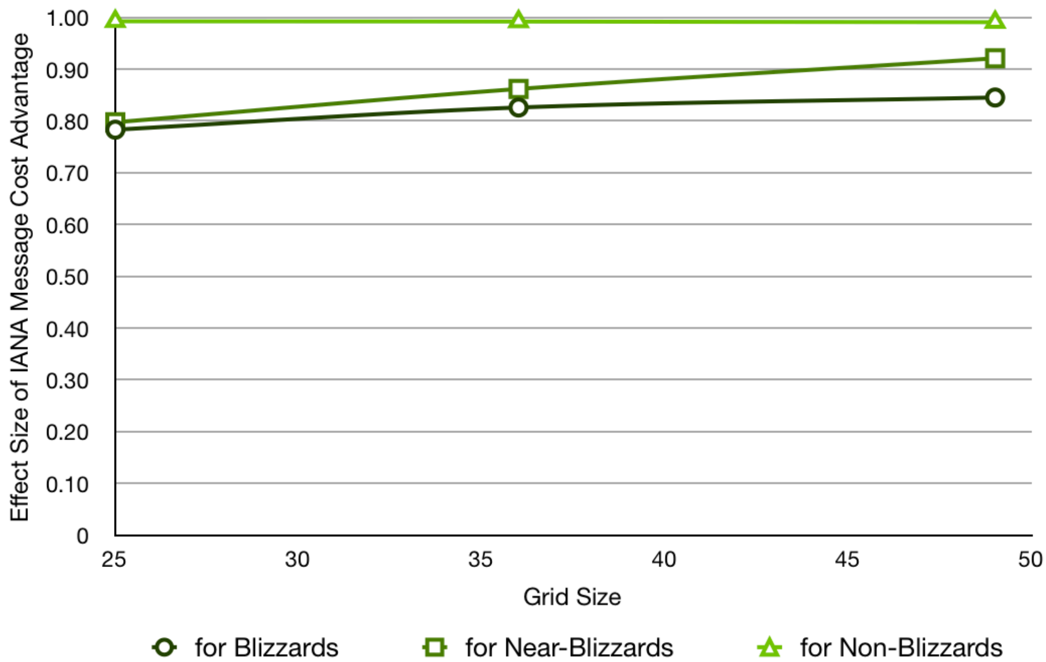

Figure 29. Effect Size of IANA Message Cost Advantage.

6.2.2 Message Amount

Figure 30 shows the means and standard deviations for message amount. As predicted, IANA generally performed worse, using more messages due to its negotiation protocol, with the exception of its performance in the non-blizzard situations, where, surprisingly, it performed slightly better than CDS.

Figure 31 shows the effect size of the lower message amounts. For blizzards and near-blizzards, CDS had the advantage. For non-blizzards, IANA had the advantage. As predicted by our hypothesis, the effect size advantages of CDS were small (0.4 or smaller). The ANOVA established these effect sizes with a p-value less than 0.01, giving them greater than 99% confidence.

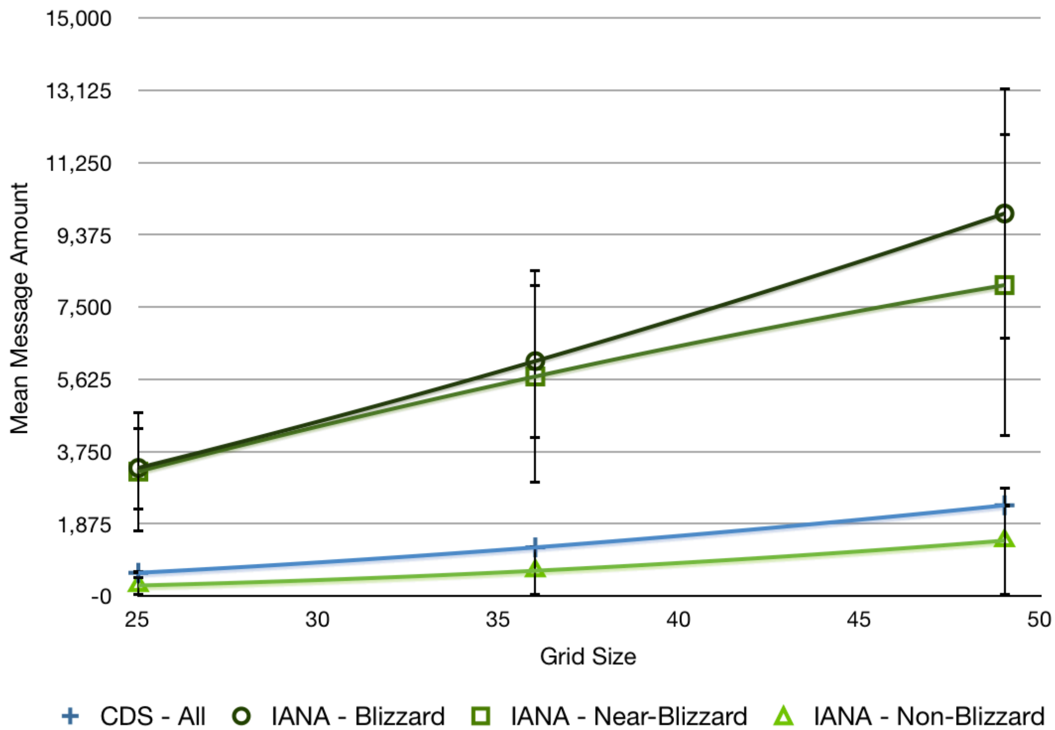

Figure 30. Message Amount Means and Standard Deviations.

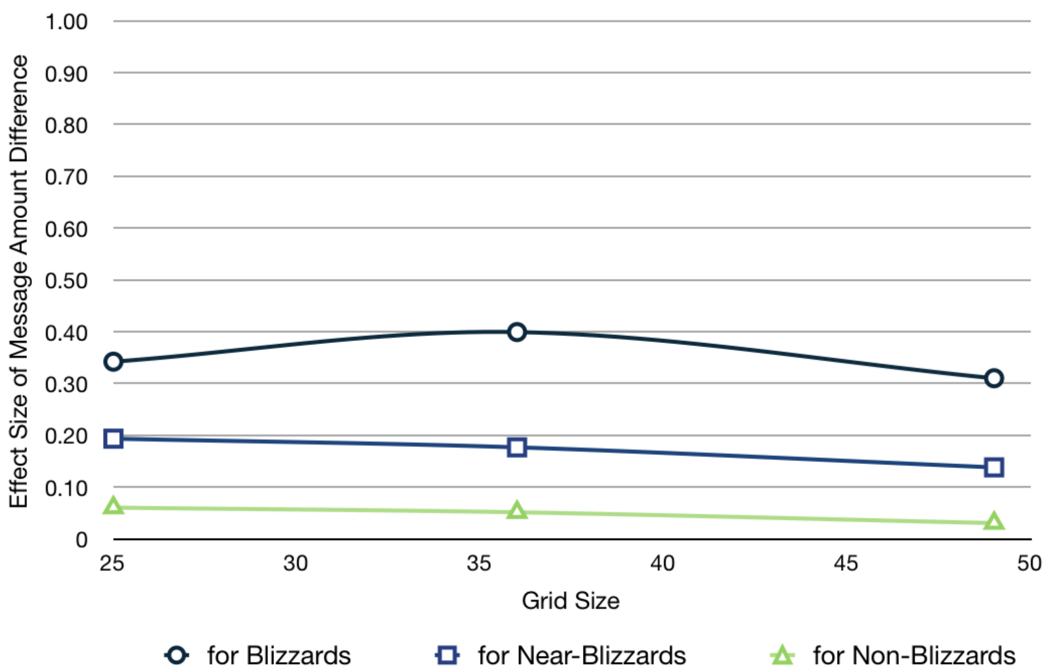

Figure 31. Effect Size of Message Amount Difference.

6.2.3 Total Accuracy

Figure 32 shows the means and standard deviations for total accuracy. As predicted, CDS performed at 100% accuracy for all geophysical situations with IANA usually lagging behind, with the exception of its performance on non-blizzards, which had produced nearly the same accuracy as CDS.

Figure 33 shows the effect size of the higher total accuracies that CDS had over IANA. As predicted by our hypothesis, all of these effect sizes were small (below 0.4, indeed, below 0.2). The ANOVA established these effect sizes with a p-value less than 0.01, giving them greater than 99% confidence, with the exception of the performance on non-blizzards, where confidence was only 50% because IANA scored so highly that its performance was nearly indistinguishable from CDS.

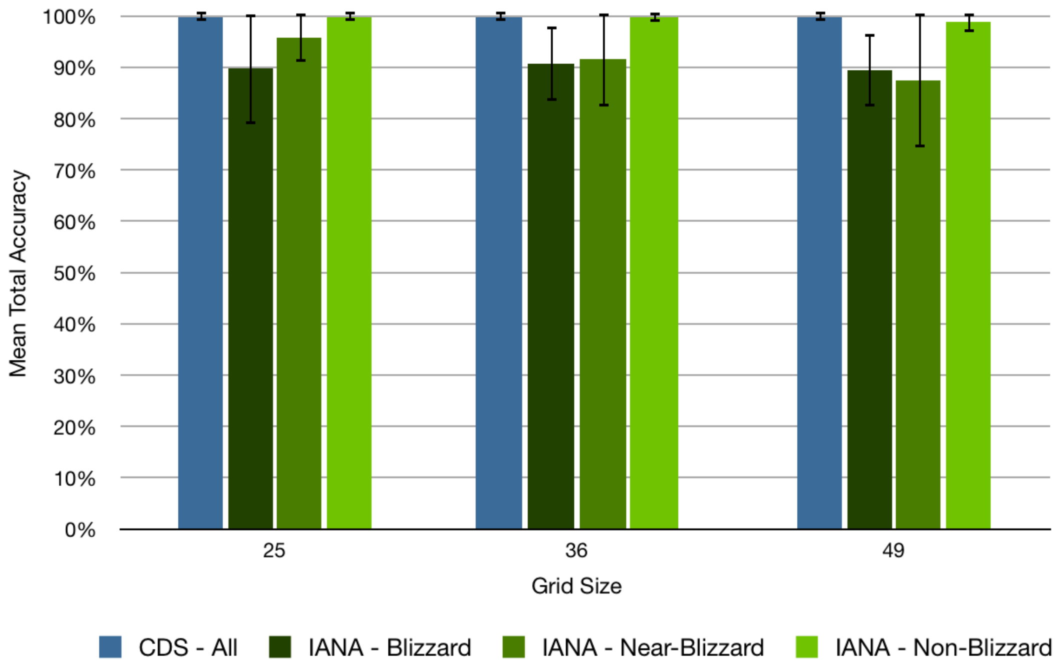

Figure 32. Total Accuracy Means and Standard Deviations.

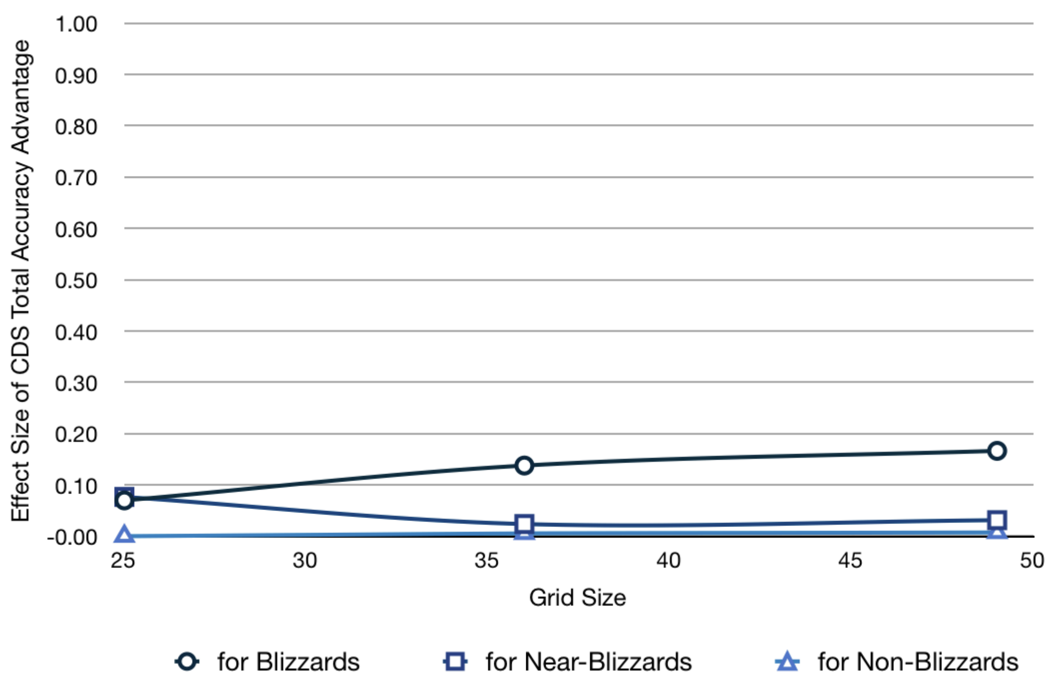

Figure 33. Effect Size of CDS Total Accuracy Advantage.

6.2.4 Super Region Accuracy

Figure 34 shows the means and standard deviations for super region accuracy. As predicted, CDS performed at 100% accuracy for all geophysical situations with IANA usually lagging behind.

Figure 35 shows the effect size of the higher super region accuracies that CDS had over IANA. As predicted by our hypothesis, all of these effect sizes were small (below 0.4, indeed, below 0.1). The ANOVA established these effect sizes with a p-value less than 0.01, giving them greater than 99% confidence, with the exception of the performance on non-blizzards, where IANA again scored so highly that its performance was nearly indistinguishable from CDS.

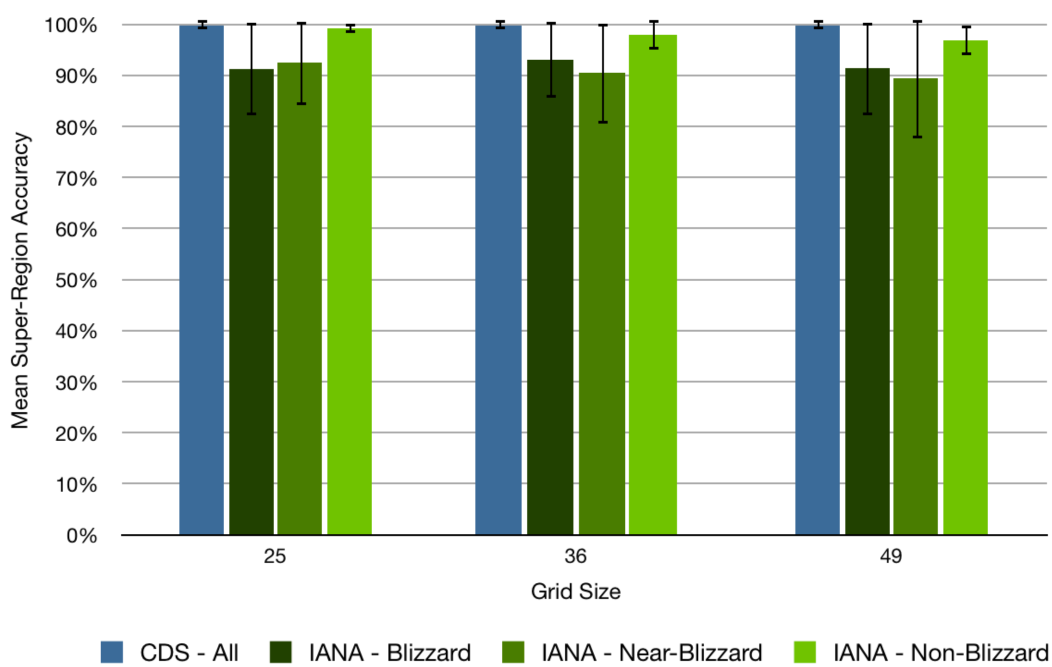

Figure 34. Super Region Means and Standard Deviations.

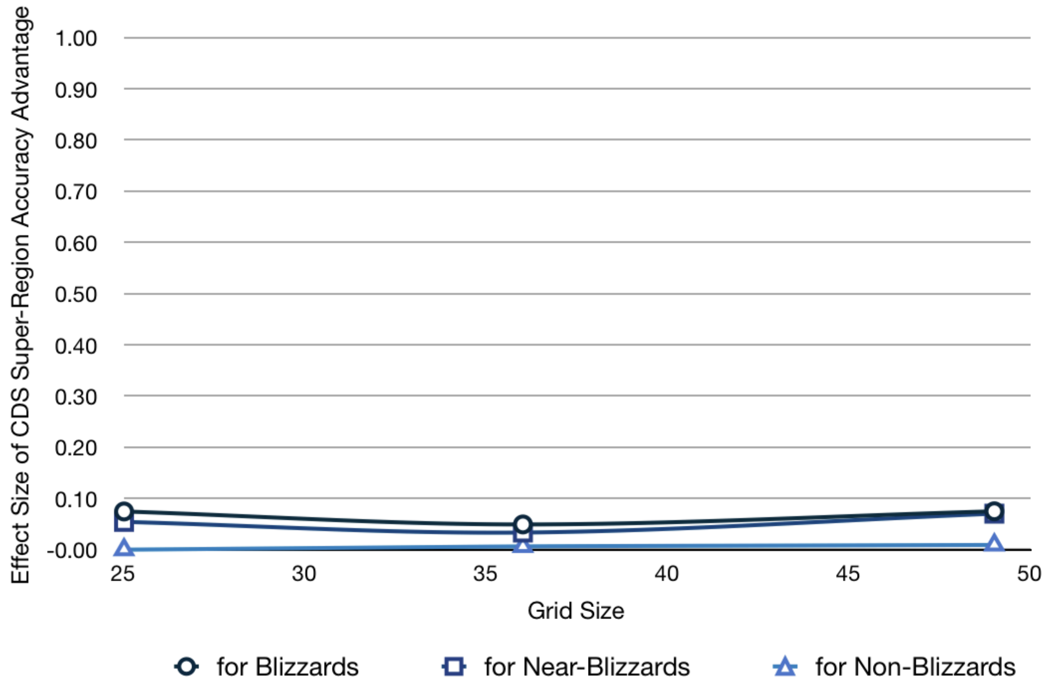

Figure 35. Effect Size of CDS Super Region Accuracy Advantage.

6.2.5 Super Strength Accuracy

Figure 36 shows the means and standard deviations for super strength accuracy. As predicted, CDS performed at 100% accuracy for all geophysical situations with IANA usually lagging behind.

Figure 37 shows the effect size of the higher super strength accuracies that CDS had over IANA. As predicted by our hypothesis, all of these effect sizes were small (below 0.4). The ANOVA established these effect sizes with a p-value less than 0.01, giving them greater than 99% confidence, with the exception of the performance on non-blizzards, where IANA again scored so highly that its performance was nearly indistinguishable from CDS.

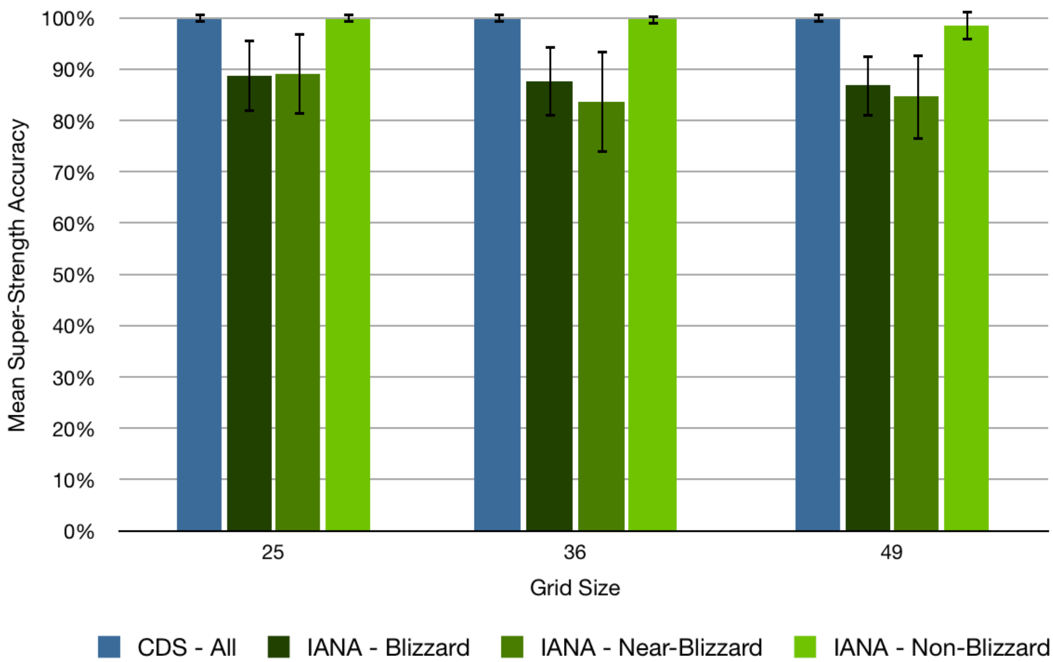

Figure 36. Super Strength Accuracy Means and Standard Deviations.

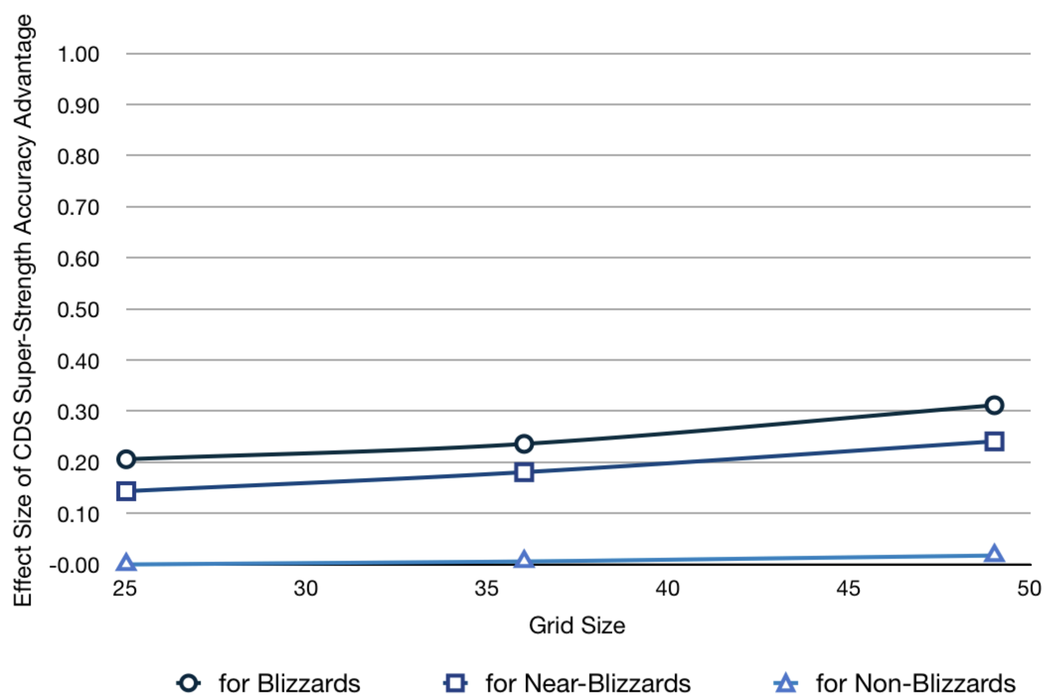

Figure 37. Effect Size of CDS Super Strength Accuracy Advantage.

6.3 Summary

As hypothesized, CDS was found to have a *small* effect size advantage in terms of accuracy and message amount, while IANA had a *large* effect size advantage in terms of message cost. This means that unless considerably higher message cost is acceptable to achieve maximal accuracy and minimal message amount, it is more suitable to use IANA than CDS, due to its *nearly* maximal accuracy and *nearly* minimal message amount and *considerably* lower message cost. A lower message cost means less time spent broadcasting messages which in turn means less power consumption. In other words, CDS is marginally more accurate and uses marginally fewer messages but has a huge cost in terms of total message size.

7 Conclusion Negotiation Only

The **Conclusion Negotiation Only** (CNO) system shares only high-level conclusions produced from individual agent models between agents. The agents then combine their conclusions using Kalman Filtering. It is our hypothesis that CNO will perform better than IANA for all communication metrics. We think CNO will have fewer **message costs** and lower **message amount** because, intuitively, its protocol requires fewer messages and smaller messages. It is also our hypothesis that IANA will perform better than CNO for the **accuracy metrics** and that this advantage will have a *large* effect size.

If these hypotheses are true, then one can argue that, unless minimal communication costs are required by a system and lower accuracy is acceptable, it is more suitable to use IANA than CNO, due to the strong advantages given by the higher accuracy of IANA. The descriptions and experiments in this chapter first appeared in Redford and Agah (2012).

7.1 Design

The CDS system presented a logical competitor for IANA. It shared all observations and always attained full accuracy in comparison to the evaluator agent's model but with significantly higher message costs. For the CNO competitor, a literature survey was done to search for published strategies in the problem domain of Decentralized

Sensor Fusion.

7.1.1 Searching for a Competitor

In order to qualify as a competitor, the candidate system needed to be able to (1) use quantifiable observations to reach conclusions and (2) produce a detailed set of (R, C, S) tuples as produced by the conclusions of field and phenomenon arguments in the collections used by IANA agents employing EFA. Condition 2 is necessary because that is the information by which the evaluator agent judges the accuracy of the agent system's models.

The multi-agent Sensor Fusion system used in Pavlin *et al.* (2010) at first appears relevant as a competitor. Similar to IANA, the system is dedicated to the Decentralized Sensor Fusion problem domain. Also like IANA, the system is used to create an accurate model of a sensing situation distributed across a group of agents attached to different sensors. An example used in Pavlin *et al.* (2010) is gas detection and an example in the related approach of Pavlin *et al.* (2006) is fire detection. Finally, the inference trees they use have a distinct resemblance to the inference trees used in the framework of Prakken (2010) on which IANA's EFA is formally based. However, the Bayesian approach they employ limits the applicability of their approach in an important way: it cannot meet condition 2 specified above. Continuous scalar values cannot be produced by the Bayesian approach they employ: only probability values can. Technically, boolean or probabilistic values can be used to represent scalar values and, indeed, boolean values are used to do so inside of computing systems. But the approach in Pavlin *et al.* (2010) does not directly implement such a conversion and, as a result, the system appears to be unable to meet condition 2 at the present time.

In addition to Bayesian approaches, another popular Sensor Fusion approach is the application of Dempster-Shafer evidence theory, such as the ones employed by Basir *et al.* (2007) or Hong *et al.* (2009). The approach of Hong *et al.* (2009) in particular is applied to the situational sensing problem of sensing events in a "Smart Home". Also, Dempster-Shafer applications employ inference trees, just as EFA and Bayesian approaches do. However, like the Bayesian approaches, the Dempster-Shafer approaches, as employed by these papers, cannot directly produce continuous scalar values or data structures that represent detailed models. So they similarly cannot meet condition 2.

A type of Sensor Fusion approach that *can* process and produce continuous scalar data, allowing it to meet both conditions 1 and 2, is Distributed Kalman Filtering (Gan *et al*., 2001; Olfati-Saber, 2005; Olfati-Saber, 2007). Due to its ability to process continuous data, Distributed Kalman Filtering (DKF), appears to be the most suitable approach of those surveyed. However, the reviewed approaches are all applied to the Sensor Fusion problem of tracking a moving target. This problem is different than situation modeling problem solved by IANA. Therefore, the methods of the reviewed approaches could not be directly applied as competitors and it was necessary to develop a new Kalman Filtering system for the situation modeling problem.

7.1.2 Building a Competitor

Kalman Filters are employed by sensors in Decentralized Sensor Fusion applications to reach consensus on a set of values. For this reason, a reasonable application for Kalman Filters is in reaching consensus on the situation summary tuple (P, F, S, R). This meets the goal of having lower communication costs than IANA, since agents in the system will only exchange situation summaries and not observations. Since only the conclusions of the agents are negotiated in this agent system, summarized in the form of the situation summary, it will be called the Conclusion Negotiating Only (CNO) system.

Like IANA, this approach also has two stages. The first stage will be the same as that used by IANA: the agents read their three local observations. To avoid having to create an entirely new inference architecture, the sensor agents in CNO are granted an EFA argument collection data structure in order to build their initial conclusions. If the sensor agent's granted collection results in the formation of a new field argument, it enters the second stage of communication. If not, it waits for communications from sensor agents who did form new field arguments, as is done in IANA.

In the second stage, eligible sensor agents send their situation summary to all other agents as a single message. This requires fewer messages and less communication than IANA since situation summaries are smaller than observations. Each sensor agent in the system takes its list of (P, F, S, R) situation summary tuples and applies a

Kalman Filter across each value of the list. This allows the agent to develop consensus values (P′, F′, S′, R′) for the situation summary or, in full detail: (P′, (Wc′, Ws′, V′), S′, R′). From these consensus values, the sensor agent infers what regions and strengths should be defined as estimations for each of the (R, C, S) tuples required by the evaluator using geometric calculation algorithms. For example, if the consensus value for P′ is 2 and the agent only has one (R, C, S) tuple representing a phenomenon, it estimates the values of a new (R, C, S) tuple distributing unused values of S′, the consensus super strength, and keeping within the bounds of R′, the consensus super region. If, on the other hand, the consensus value for P′ is 2 and the agent has three (R, C, S) tuples representing phenomena, it combines the regions and strengths of two of the tuples into a single S value and R value (the value of C will be the same). Values from F′ are used to make the same types of geometric inferences for their (R, C, S) tuples for each individual type of field in (Wc′, Ws′, V′).

After all the sensor agents have completed their Kalman Filter calculations and (R, C, S) tuple estimates, they send their final set of (R, C, S) tuples to the evaluator. The evaluator then evaluates the performance of the agent system based on the overall performance of all agents in the system.

Initial testing of this approach found that the accuracy of the CNO system was unacceptably low if zero communication was allowed between the sensor agents before forming their situation summaries. Intuitively, this makes sense. If a sensor agent is allowed to summarize the entire sensing situation using only their local observations, they are summarizing based on sensor readings in a single geographical location. Applying a Kalman Filter to a set of such summaries would lead to an unacceptable consensus, especially for the value of R, the super region, which is derived from location based data. Under such circumstances, the only value of R that each sensor agent could have is a circle around its own location.

Due to the low accuracy performance that results from using zero observation communication, some minimal observation sharing is incorporated between the first and second stages of communication. Specifically, after reading its three observations, if a CNO sensor agent has two or more observations that can be used to build a field argument, it randomly selects one of those observations and sends it. If the CNO sensor agent has a single relevant observation, it sends that observation if both $X \equiv 0 \pmod 2$ and $Y \equiv 0 \pmod 2$ where (X, Y) are the value of its sensor's geographical coordinate. These specific calculations were used because they allowed CNO to have lower communication costs than IANA in preliminary testing while still having comparable accuracy. Similarly evenly distributed calculations could be used if the *in-situ* sensors were not on a square grid. After this minimal sharing stage is complete, CNO agents enter the second stage of Kalman Filtering on the situation summary tuples as explained. It is still valid to call this approach Conclusion Negotiating Only because only the conclusions of situation summary tuples are negotiated by the agents' Kalman Filters.

7.2 Random Observation Priming

Based on preliminary testing, the CNO approach performs best if agents in the system prime themselves with some random observations from across the *in-situ* sensor grid. This allows the CNO approach to better compete with IANA and is an acceptable precondition so long as IANA is given the same advantage. Therefore, for the experimental setup, it is assumed that all agents in both systems first randomly exchange some observations to prime their knowledge bases prior to executing their respective communication protocols. All evaluations are done after this random priming takes place.

While this random priming is helpful, the percentage of observations shared before evaluation should be as minimal as possible to allow CNO to strongly compete. Otherwise, the differences between IANA and CNO will be difficult to distinguish from the random priming. To find an acceptable percentage of random observations, some preliminary comparisons were made between the performance of CNO and the performance of a system that only performed the random priming. For these preliminary comparisons, this system is referred to as Random Only. It is unnecessary to include IANA in these comparisons because, as shown in the Complete Data Sharing chapter, IANA can achieve 80% or higher accuracy without random observation priming.

For these preliminary comparisons, 10 Blizzard scenarios from the data from Wolfram Mathematica (2011) were tested for both CNO and Random using square grids of 49 agents. This combination of scenarios and agent grids were tested for random priming percentages from 0% to 100%. So, for 0%, the agents exchanged 0% of random observations prior to executing their situation modeling protocol (or just doing nothing in the case of Random Only). And for 10%, they exchanged 10% of random observations prior to executing, etc. Figure 38 shows the means and standard deviations for the two systems for each random priming percentage. As it can be seen, the advantage of CNO versus Random is most pronounced for the lower percentages of random observations primed. Intuitively, this makes sense because, along with its communication protocol and Kalman Filtering, CNO should infer much more accurate conclusions than a system that just exchanged some random observations between sensors with no additional communication or processing. A good percentage of random observations to grant both CNO and IANA appears to be 30% observations given. This allows CNO to attain greater than 75% accuracy, which is intuitively "average". Allowing more random observations to be primed makes CNO less distinguishable in performance from Random Only, which is undesirable.

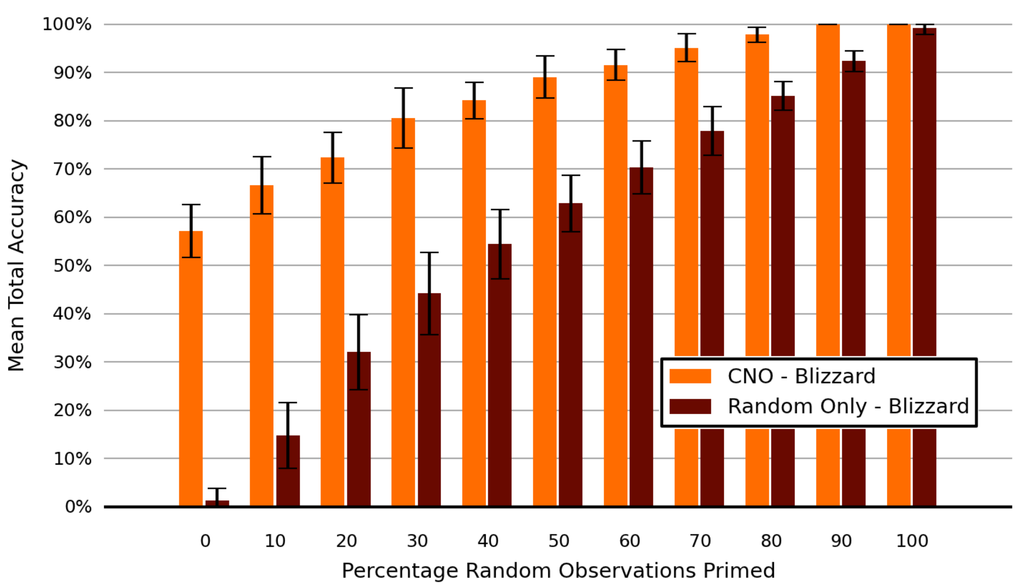

Figure 38. Mean Total Accuracy for increasing random priming observation

percentages.

The effect size advantage of CNO over Random Only in Figure 38 can be seen in Figure 39. This reveals that all advantages for CNO can be classified as “high”, except for 100% observations given, which results in such similarly high accuracies that CNO only has a “low” effect size over Random. Since effect size is “high” for all other values, we will select 30% observations primed to be used for all other experiments in this paper since that is the minimum amount that gives CNO greater than 75% accuracy.

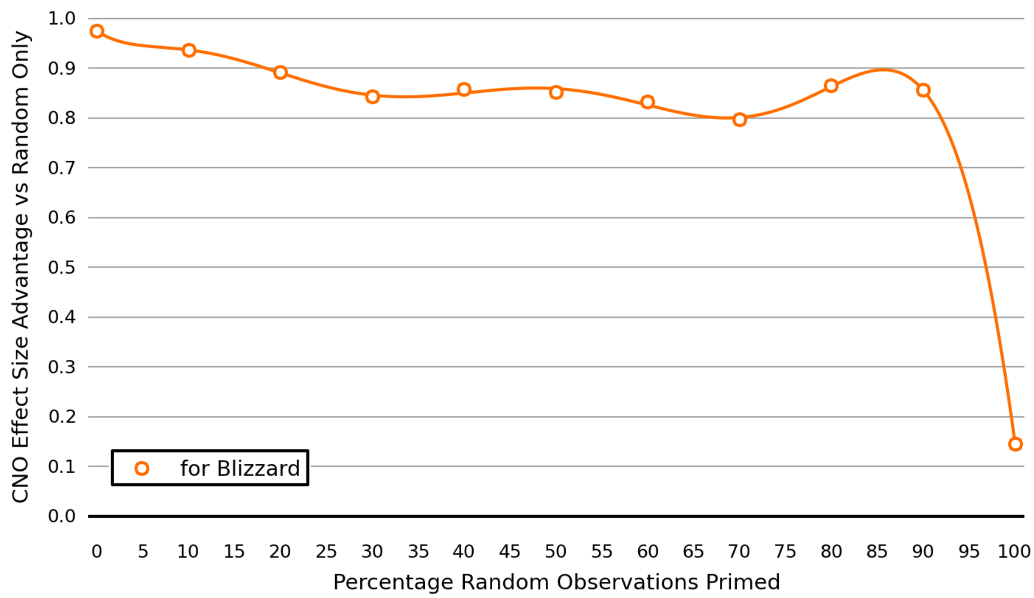

Figure 39. Effect Size of IANA Message Cost Advantage.

7.3 Comparison

Given the results of the preliminary comparisons in the previous section both CNO and IANA will prime their systems by exchanging 30% of random observations across the entire *in-situ* sensor agent grid prior to the full metric experiments for all scenarios.

For the metric experiments, three different types of historical real-world scenario data were taken from Wolfram Mathematica (2011): Blizzards, Near-Blizzards, and Non-Blizzards. In Blizzard scenarios, a definite and large overlap of high wind speed, low wind chill, and low visibility values, constituting one or more large blizzards, exists on the map. In Near-Blizzard scenarios, large patches of such values exist but do not always overlap to form blizzards. In Non-Blizzard scenarios, almost no patches of such values exist and blizzards are never present. For these experiments, 30 Blizzards, 30 Near-Blizzards, and 30 Near-Blizzards were selected from real-world weather situations from Russia, China, and the United States using historical weather data from 2000 to 2010. For each real-world scenario, both CNO's and IANA's agents use the same pseudo-random number generator seeds to ensure they prime themselves with exactly the same random observations from that scenario.

To test scalability, each of these 90 scenarios were tested for three different *in-situ* Sensor Web grid sizes: 25 sensor stations, 36 sensor stations, and 49 sensor stations. As mentioned earlier, each sensor station is associated with a single agent. These grid sizes were chosen because they represent perfect square grids of increasing size. This results in a total of 270 test situations.

For each of the 270 test situations, only a single CNO experiment is performed. This is because, given a particular scenario, CNO's sensor agents will always make the same local observations and, given the same random seed for priming, they will always start with the same random observations for a scenario. Thus, given a particular scenario and random seed, CNO's sensor agents will always exchange the same scenario summaries and Kalman Filtering on those summaries will always produce the same results.

However, for each of the 270 test situations, 10 IANA experiments are performed. This is because sensor agents in IANA may produce different conclusions depending on what order the agents in the system execute the SoD protocol. This can be explained with the following hypothetical scenario. An agent A starts with all three of its local observations as relevant and agent B starts with zero of its local observations as relevant. If, at the start of the simulation, A is the first agent that contacts agent B, the agents are guaranteed to disagree because A will have one of each type of field argument and B will have zero field arguments. Therefore, A will send the relatively large message containing all three of its observations. However, if other agents near A had contacted B first, it is likely that they will agree and the observation exchange will not occur. Depending on the combination of agent communications, more or less communication may have taken place. Also, more or less premature agreements between agents may have taken place, resulting in more or less accuracy.

With the experiments performed for the 270 CNO tests and and 2700 IANA tests, the comparative performance of the two systems will now be analyzed using each metric.

7.3.1 Message Cost

Figure 40 shows the means and standard deviations over all the experiments for message cost. As predicted, for each individual type of scenario, CNO performs better than IANA. For Blizzards (represented by a circle), CNO's message cost was directly below IANA's. The same pattern holds for Near-Blizzards and Non-Blizzards: for each type of scenario, CNO's performance was better than IANA's. The SoD protocol used by IANA uses more communication than CNO's simple situation summary exchanges (along with a few single observation exchanges).

However, also as predicted, the omega-squared effect size of this lower communication cost advantage is very low, far below 0.1 and close to 0.0, as can be seen in Figure 41. This means that, from the perspective of omega-squared effect size, CNO's victory was extremely small, an almost imperceptible advantage.

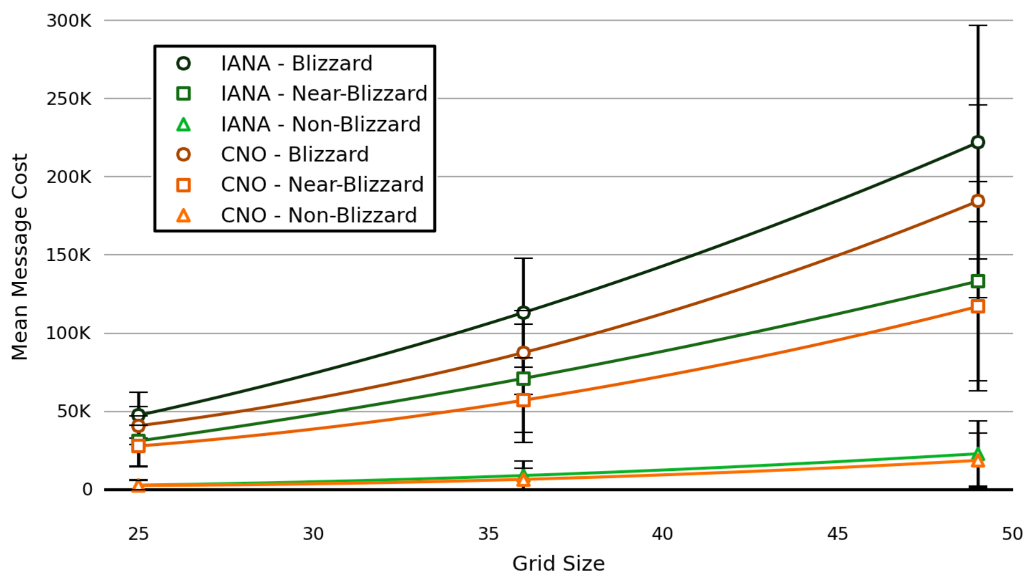

Figure 40. Message Cost Means and Standard Deviations.

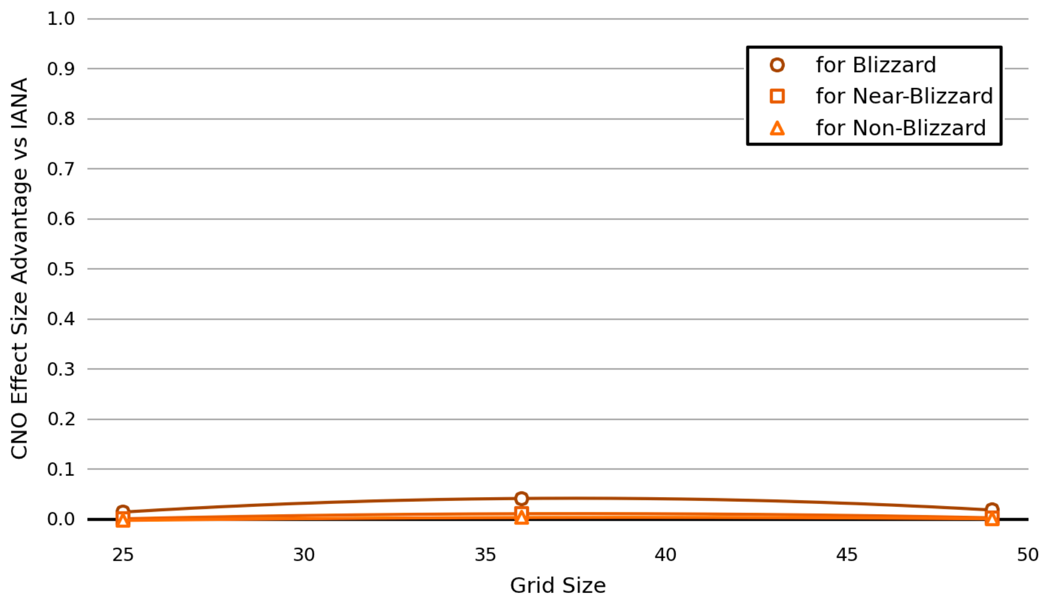

Figure 41. Effect Size of CNO Message Cost Advantage.

7.3.2 Message Amount

Figure 42 shows the means and standard deviations for message amount. As predicted, IANA performed worse, using more messages due to its SoD protocol, which exchanges many small messages to verify agreement or disagreement.

Figure 43 shows the omega-squared effect size of CNO's lower message amount advantage. As predicted by our hypothesis, the effect size for CNO on Near-Blizzards and Non-Blizzards were undeniably small (0.4 or smaller). However for Blizzards, the results were not as predicted by our hypothesis. The effect size was very close to 0.4, and in the case of the 35 sensor grid, higher than 0.4, making the effect size advantage for Blizzards barely medium. This means that, if low message amount is absolutely necessary for an application, and a high amount of sensing situations are expected (Blizzard scenarios in this case), there is a medium effect-size advantage to using CNO's situation summary exchange combined with Kalman Filtering approach.

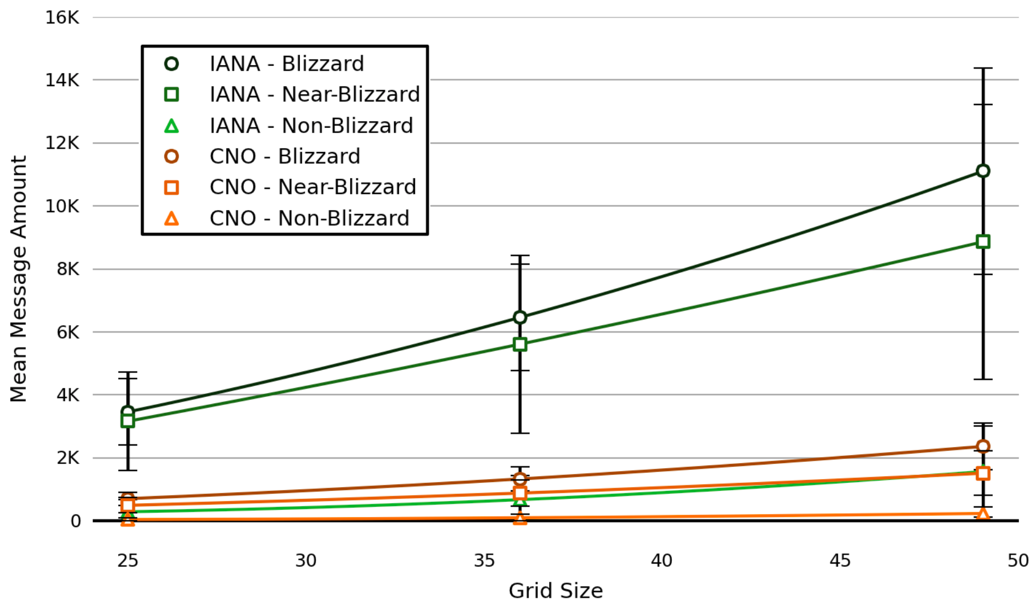

Figure 42. Message Amount Means and Standard Deviations.

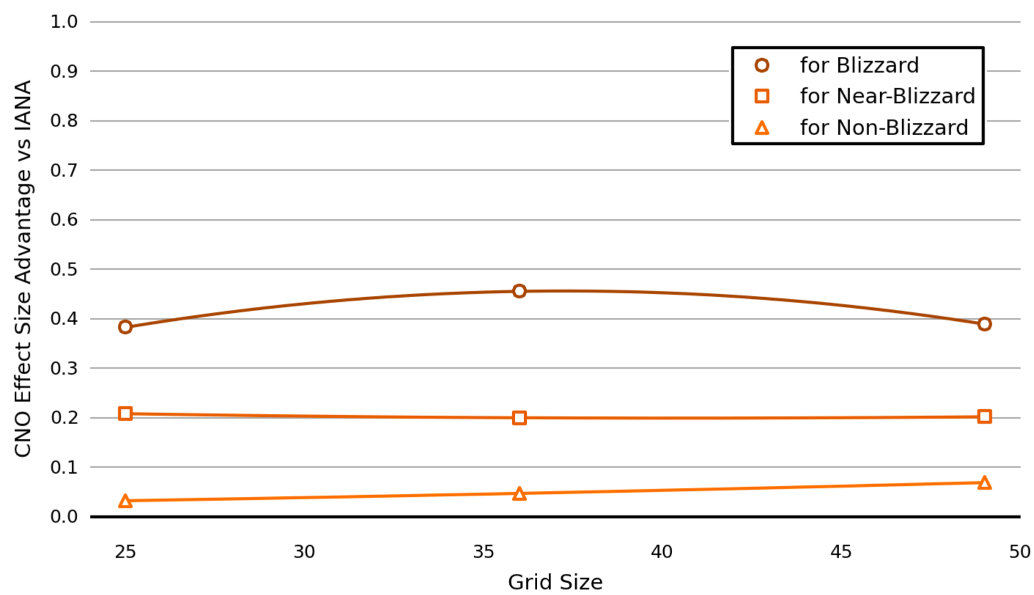

Figure 43. Effect Size of CNO Message Amount Advantage.

7.3.3 Total Accuracy

Figure 44 shows the means and standard deviations for mean total accuracy. As predicted, IANA had a higher accuracy for all sensing scenarios. In addition, its accuracy was always over 90% and close to 100% in many cases. CNO was never quite able to achieve mean accuracy over 90% and was typically closer to 70%.

Figure 45 shows the effect size of the higher total accuracies that IANA had over CNO. As predicted by our hypothesis, the effects size advantage for IANA for Blizzards and Near-Blizzards was always large or very close to large (above 0.8 or very close to 0.8). This was also true for Non-Blizzards where the sensor grid size was 49. However, for Non-Blizzards where the grid size was 25 or 49, the results were not as predicted by our hypothesis. For the agent grid size of 25, the effect size advantage of IANA was only small and for the agent grid size of 36, the advantage was only medium.

Referring back to Figure 44 and comparing the two systems for Non-Blizzards for grid sizes 25 and 36 reveals why this is the case. For both situations, the standard deviations were unusually high. The accuracy for CNO varied largely in these situations, from about 70% to 100% and from about 55% to 100%, respectively.

Intuitively, this makes sense because, for the Non-Blizzard scenario, very few observations meet any kind of relevant conditions related to fields or phenomena. So it is understandable that, in these scenarios, the CNO system could infer very badly or very well, depending on whether it was lucky enough to reach the correct Kalman Filtering consensus on all of the very few relevant areas of interest or only about 55 to 70 percent of them.

It is noteworthy that, despite these two anomalous low effect sizes: for the tested scenarios, IANA *always* has a higher mean accuracy than CNO, *almost always* with a large or nearly large effect size, and *always* has a mean accuracy higher than 90%. For these reasons, despite the two anomalous low effect sizes, IANA is still recommended over CNO if high detailed model accuracy has any significance to an application. It is also noteworthy that the scenarios that these low effect sizes occurred for were the Non-Blizzard scenarios, which are scenarios where only a low amount of sensing situations are occurring.

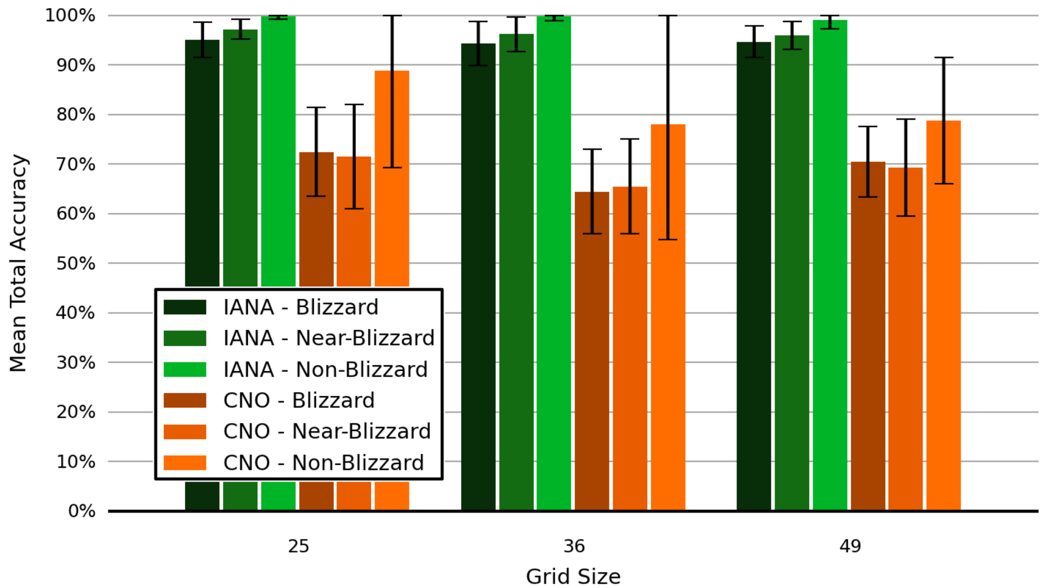

Figure 44. Total Accuracy Means and Standard Deviations.

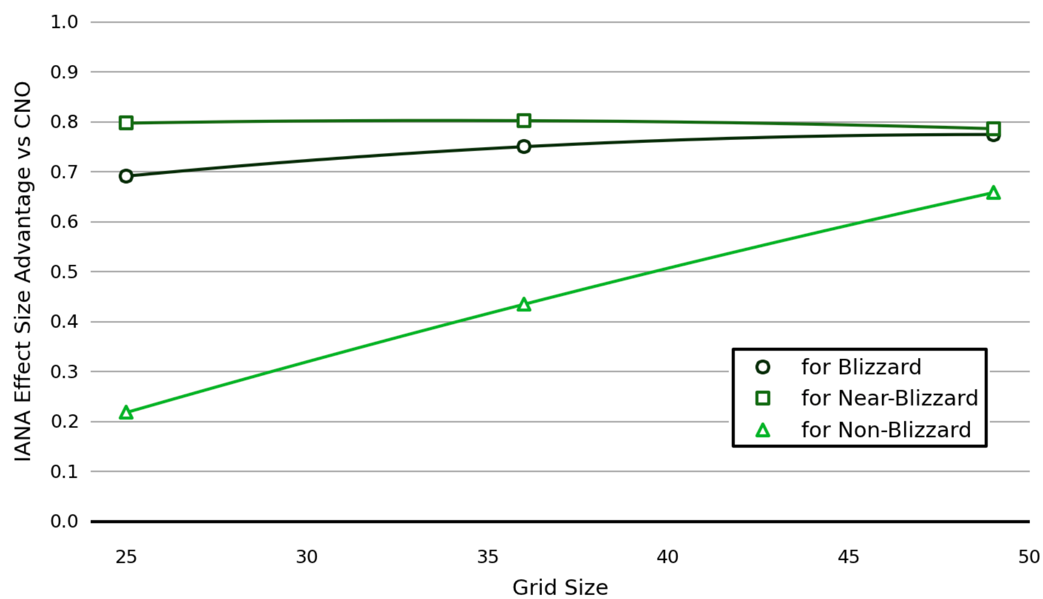

Figure 45. Effect Size of IANA Total Accuracy Advantage.

7.3.4 Super Region Accuracy

Figure 46 shows the means and standard deviations for super region accuracy. As predicted, IANA had higher accuracy for all sensing scenarios.

Figure 47 shows the effect size of the higher super region accuracies that IANA had over CNO. Contrary to our hypothesis, the super region accuracy for IANA did not have a high effect size advantage over CNO. In fact, the effect size advantage for all scenarios and grid sizes was close to medium (about 0.4).

Combining this result with the *message amount* effect size results from Figure 31 allows an interesting conclusion. If an application meets the following conditions, it may be reasonable to use CNO rather than IANA:

1) All that is needed is a reported super region of overall high sensing activity (i.e., a detailed account of the sensing situation, as provided by *total accuracy*, is *not* required).
2) High accuracy for this super region *is not* absolutely necessary.
3) Low message amount *is* absolutely necessary.
4) A high amount of sensing activity (e.g., as would happen in a Blizzard scenario) is expected. If an application does not meet all of these special conditions, IANA should be preferred, due to its large effect size accuracy advantages and small effect size communication disadvantages under all other tested circumstances.

It should be noted that the standard deviations in Figure 46 for CNO are significantly larger for CNO than for IANA. Due to this, if consistency in super region accuracy is important to an application, IANA should be preferred. Finally, despite only having a medium effect size advantage over CNO, IANA still has *higher* super region accuracy than CNO, so if high super region accuracy is important for an application, IANA should be preferred.

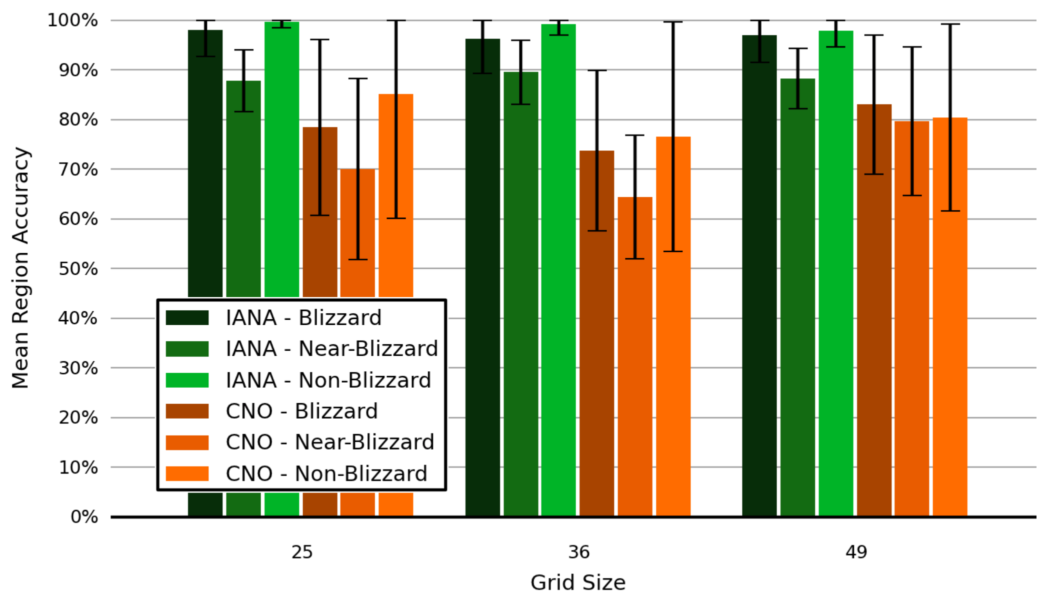

Figure 46. Super Region Means and Standard Deviations.

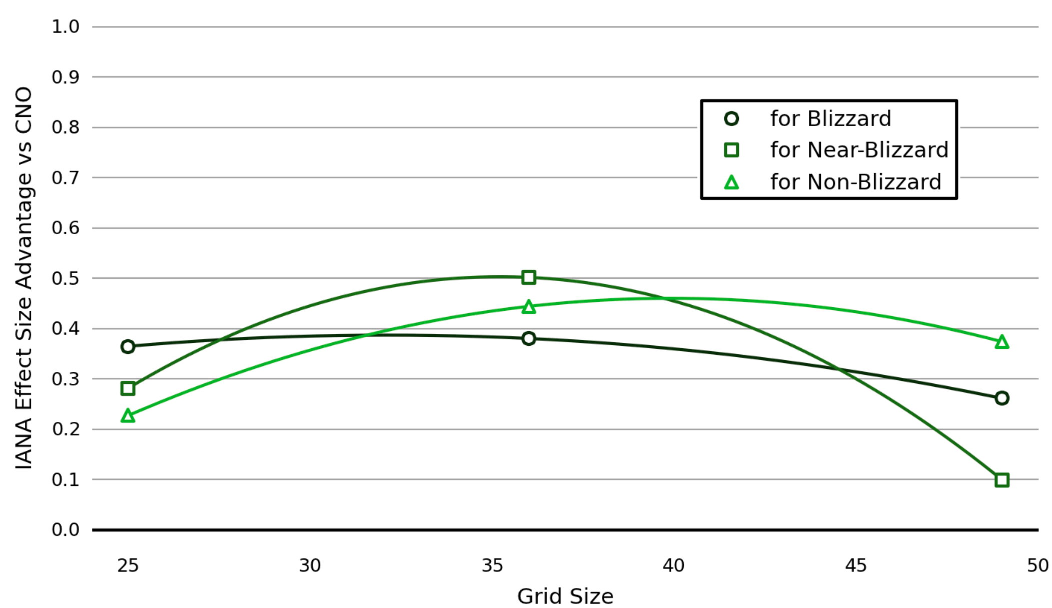

Figure 47. Effect Size of IANA Super Region Accuracy Advantage.

7.3.5 Super Strength Accuracy

Figure 48 shows the means and standard deviations for super strength accuracy. As predicted, IANA had higher accuracy for all sensing scenarios.

Figure 49 shows the effect size of the higher super strength accuracies that IANA had over CNO. As predicted by our hypothesis, the effects size advantage for IANA for Blizzards where the sensor grid size was 36 or 49 was very close to large (close to 0.8). Contrary to our hypothesis, the super strength accuracy for IANA did not have a high effect size advantage over CNO for any other scenarios. For grid size 36 and 49 on Near-Blizzards, and grid size 25 for Blizzards, the effect size was medium. For all other scenarios, the effect size was small. Also, unlike the super region accuracies for CNO, the standard deviations in Figure 48 are comparable to (though still generally larger than) those of IANA.

However, it should be noted that all of IANA's mean super region accuracies are higher than 85% and, with the exception of Non-Blizzards, CNO's accuracies are all about 75% or lower. For Non-Blizzards, IANA's accuracies are still higher than CNO's but CNO's all indeed average at 90% or above.

Some reflection on the nature of the super strength metric is relevant as well. The super strength metric is a sum total of relevant observations in the scenario (observations that can be used to build arguments for blizzards or their field sub-arguments). As such, the super strength metric is only a single value. For a sensor grid of size 25, the highest possible super strength value is 75, since at most 3 observations per sensor are relevant. For a sensor grid of size 49, the highest possible value is 147.

As observed earlier, the scenario type that CNO performed best on in comparison to IANA was Non-Blizzard. We have referred back to the data the evaluator used to judge super strength in the Non-Blizzard scenarios. The highest possible super strength value in these scenarios for the sensor grid size of 49 was 14. The value was lower for smaller sensor grid sizes and often zero. For comparison, in the Near-Blizzard scenarios, the highest super strength value for the sensor grid size of 49 was 48. The value was usually 20 or higher for a grid size of 49 and 10 or higher for other grid sizes.

As the Bayesian and Dempster-Shafer approaches to Sensor Fusion demonstrate, sometimes detection versus non-detection is an important issue in-situational modeling (Prakken, 2010; Hong *et al.*, 2009). But it should be noted that this is basically the type of situational modeling that the Non-Blizzard scenario begins to reduce to for the metric of super strength. As opposed to the Blizzard and Near-Blizzard scenarios, for which super strength modelling is actually an issue of both detecting activity *and* accurately judging its magnitude.

In summary, this was still a victory for IANA but, for Near-Blizzards and Non-Blizzards, not as strong a victory as predicted.

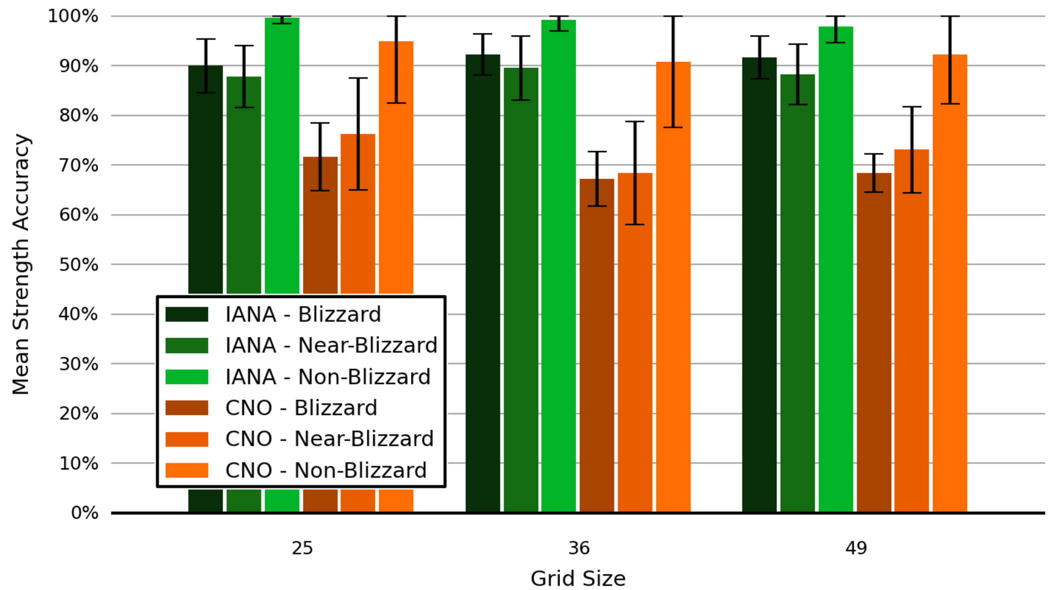

Figure 48. Super Strength Accuracy Means and Standard Deviations.

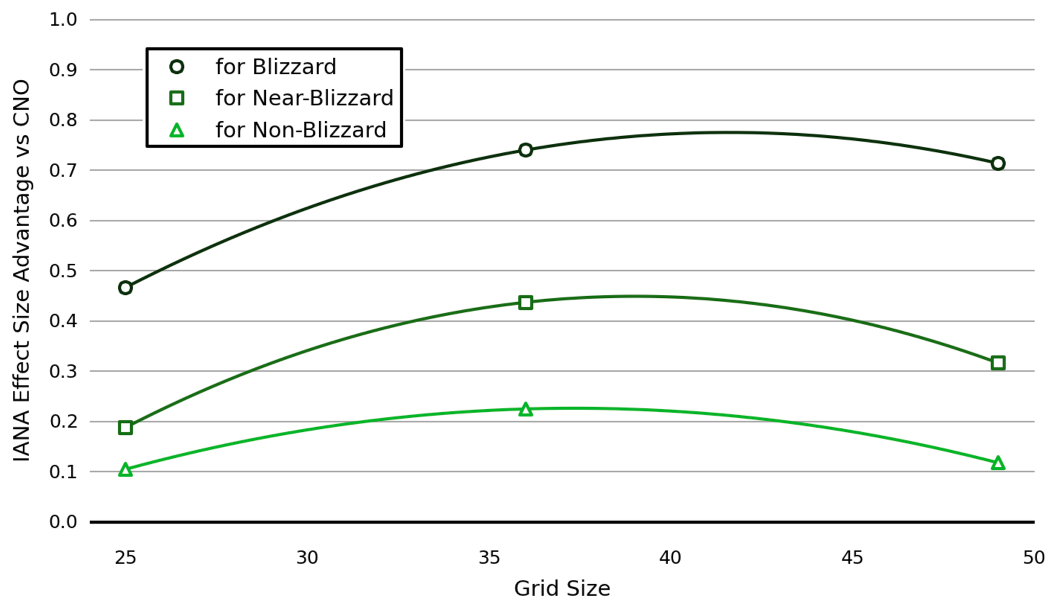

Figure 49. Effect Size of IANA Super Strength Accuracy Advantage.

7.4 Summary

Matching closely to the hypothesis, IANA was found to outperform the Kalman

Filtering system in terms of total accuracy with mostly *large* and *medium* effect sizes. To summarize the results of the CNO experiments: if low message amount is absolutely necessary and a high amount of sensing activity is expected, then the Kalman Filtering approach used had acceptable accuracy (though still lower accuracy than IANA) when used for overall general values like super region and super strength. However, if the primary concern is message cost, then IANA is clearly the better choice, especially if a highly detailed and accurate model of the situation is required. It should also be noted that *random observation priming* was necessary prior to the execution of the Kalman Filtering approach to achieve *acceptable* (~70%) accuracy results. As shown in the Complete Data Sharing chapter, this priming is *not* necessary for IANA using the SoD protocol to achieve *high* (~90%) accuracy results. And *with* priming, IANA achieved nearly perfect (~95%) accuracy results.

8 Conclusion

8.1 Contributions

This dissertation makes multiple *theoretical* contributions. Its theoretical contributions are primarily in the field of Computational Argumentation. In the chapter Evidentialist Foundationalist Argumentation (EFA), we formally established a new type of evidence-based argumentation. This argumentation is theoretically interesting because it is based solely on sensor observations from the physical world. Its possible applications reach beyond Sensor Webs to other domains of engineering as well as the areas of law, business, medicine, politics, and education. EFA is a unique instantiation of the well-established ASPIC framework for argumentation (Prakken, 2010). This dissertation also makes theoretical contributions to Sensor Webs by providing a new type of Sensor Fusion protocol and mechanism through IANA, which was detailed in the chapter Investigative Argumentation-based Negotiating Agents. A final major theoretical contribution is that IANA provides a concrete implementation and application of EFA.

EFA can be seen as representative of the Evidentialist approach to epistemology (Conee *et al.*, 2004). EFA appears to be the first approach to Computational Argumentation explicitly based on Evidentialist epistemology. Also, since EFA has been shown to be an extension of the ASPIC framework, it inherits the satisfaction of all the rationality postulates that the framework satisfies in Prakken (2010). Some specific contributions of EFA are the properties of *quantifiable* and *verifiable* evidence and a special type of defeasible rule called a *calculation rule* which is specifically applicable to quantified premises. Through the concept of the calculation rule, EFA formalizes a way to calculate conclusions directly from their premises. Through Prakken's framework, EFA can be employed in conjunction with the traditional concerns of attack and defeat in the Computational Argumentation literature, using *evidence amount ordering*.

This dissertation also makes multiple *experimental* contributions are to the fields of Computational Argumentation, Sensor Webs, and Multi-Agent Systems. It provides experimental justification for the usefulness of Computational Argumentation. It provides a comparison of argumentation (IANA), complete sharing (CDS), and Kalman Filtering (CNO) approaches for Sensor Fusion. It provides an experimental demonstration of the utility of Sensor Webs. And finally, It provides an experimental demonstration of the utility of Multi-Agent Systems. These experimental contributions are found in the chapters titled Experiments, Complete Data Sharing, and Conclusion Negotiation Only.

As hypothesized, CDS was found to have a *small* effect size advantage in terms of accuracy and message amount, while IANA had a *large* effect size advantage in terms of message cost. This means that unless considerably higher message cost is acceptable to achieve maximal accuracy and minimal message amount, it is more suitable to use IANA than CDS, due to its *nearly* maximal accuracy and *nearly* minimal message amount and *considerably* lower message cost. A lower message cost means less time spent broadcasting messages which in turn means less power consumption. In other words, CDS is marginally more accurate and uses marginally fewer messages but has a huge cost in terms of total message size.

Matching closely to the hypothesis, IANA was found to outperform the Kalman Filtering system in terms of total accuracy with mostly *high* and *medium* effect sizes. To summarize the results of the CNO experiments: if low message amount is absolutely necessary and a high amount of sensing activity is expected, then the Kalman Filtering approach used had acceptable accuracy (though still lower accuracy than IANA) when used for overall general values like super region and super strength. However, if the primary concern is message cost, then IANA is clearly the better choice, especially if a highly detailed and accurate model of the situation is required. It should also be noted that *random observation priming* was necessary prior to the execution of the Kalman Filtering approach to achieve *acceptable* (~70%) accuracy results. As shown in the Complete Data Sharing chapter, this priming is *not* necessary for IANA using the SoD protocol to achieve *high* (~90%) accuracy results. And *with* priming, IANA achieved nearly perfect (~95%) accuracy results.

8.2 Limitations and Future Work

There are many ways that EFA could be further extended and investigated. There are many other types of geophysical phenomena that could be monitored using the

Sensor Fusion approach of EFA using the SoD protocol, such as volcanos (Kedar *et al.*, 2008) and earthquakes (Scott *et al.*, 2004). For such applications, EFA using the SoD protocol could be experimentally tested against other competitors in Sensor Fusion such as other approaches to Kalman Filtering using the metrics employed here or other metrics to be defined. There are also completely different types of Sensor Web modeling applications that EFA could be applied to, such as 'Smart Home' monitoring (Hong *et al.*, 2009) or engine fault detection (Basir *et al.*, 2007). Also, Sensor Webs are just one type of application for EFA. Given that it is founded on the concept of organizing quantifiable and verifiable evidence, it has applicability to a wide variety of fields, since such evidence plays some role in most scientific and engineering applications and even in business applications. For example, EFA could have direct applications to the organization and investigation of electronic health records.

Also, it is noteworthy that in our experimental implementation of EFA, the calculation rules we used assumed complete and accurate data. Calculation rules could also be developed for detecting and correcting for missing or inaccurate sensor information. The communication order used in IANA's execution of the SoD protocol was also linear in nature. It is plausible that the order of selection of communication partners could improve performance if it were based on the geometry of the expected argument conclusions. For example, for the circular regions used in this application, it may be most advantageous for agents to communicate first in a circular pattern to agents that are far away from them.

Finally, many theoretical extensions to EFA are possible. The use of the many possible specific types of calculation rules can be explored. Specific groups of calculation rule types could possibly be generalized. The relationship between calculation rules and traditional rules in argumentation can be explored. One issue of interest is what kind of isomorphism can be established between traditional rules and calculation rules, assuming the use of a knowledge base with quantifiable members. The framework could possibly be formally extended in such a way that calculation rules can be calculated by other calculation rules, a concern related to the derivation of defeasible inference rules found in Prakken *et al.* (1997). Also, the use of the evidence amount ordering preference can be explored theoretically. Because evidence-based arguments can differ dramatically in their strength based on the Gate function employed by their calculation rule, sometimes by orders of magnitude, this is related to the theoretical question of just how much stronger one argument is than another, and what practical implications this may have for the use of arguments with strengths that are orders of magnitude smaller or larger than other arguments. Another important theoretical question is when it is appropriate to *investigate* in an attempt to find evidence to *build* the strength of an evidence-based argument that currently has a low strength compared to a competing argument that currently attacks and defeats it. Also, different *types* of evidence-based sub-arguments could be explored and generalized. Here, the question of whether there are certain evidence-based arguments that are broadly employed as sub-arguments to larger evidence-based arguments across a wide variety of problem domains is relevant. This direction of inquiry is related to the many important argument pattern generalizations, for example those made in Bex *et al.* (2003) and Walton *et al.* (2008), except specifically applied to arguments built on quantifiable and verifiable evidence as defined in this dissertation.

8.2.1 Argumentation-Based Negotiation

A natural future application of EFA is Argumentation-Based Negotiation (ABN), where agents negotiate the use of a resource based on the arguments they have for using it. Using evidence amount ordering, combined with calculation rule gate functions, agents can collaboratively agree on which agent has the greater argument for utilizing a resource.

Specifically, at the beginning of the conflict leading to the need for negotiation, each agent begins with a database of arguments and a leading argument about the conflict. This is exemplified in the Sensor Web domain in Figure 50, where two agents are negotiating over the use of a wind sensor.

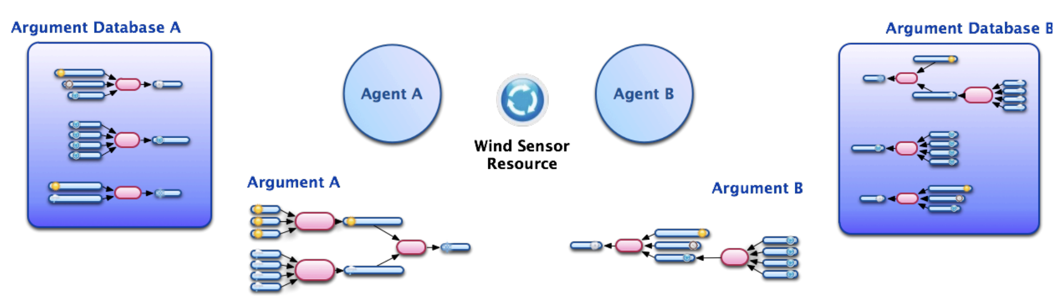

Figure 50. Each agent presents its argument for the use of the wind sensor.

The agents could just decide which agent gets to use the resource based solely on comparative argument strengths. However, this could potentially miss important opportunities. What if Agent B has evidence that would make Agent A's argument stronger or vice-versa? Or what if, upon exchanging evidence, Agent A finds an argument in its database that would make it prefer using a different resource, leading to the mutually beneficial situation where Agent A concedes and Agent B may take the resource originally in conflict? The modification of argument databases based on exchanged evidence can lead to many useful and unpredictable consequences. Therefore, it may be important that the agents extract and exchange the evidence used in their arguments to update their respective databases. This situation is shown in Figure 51.

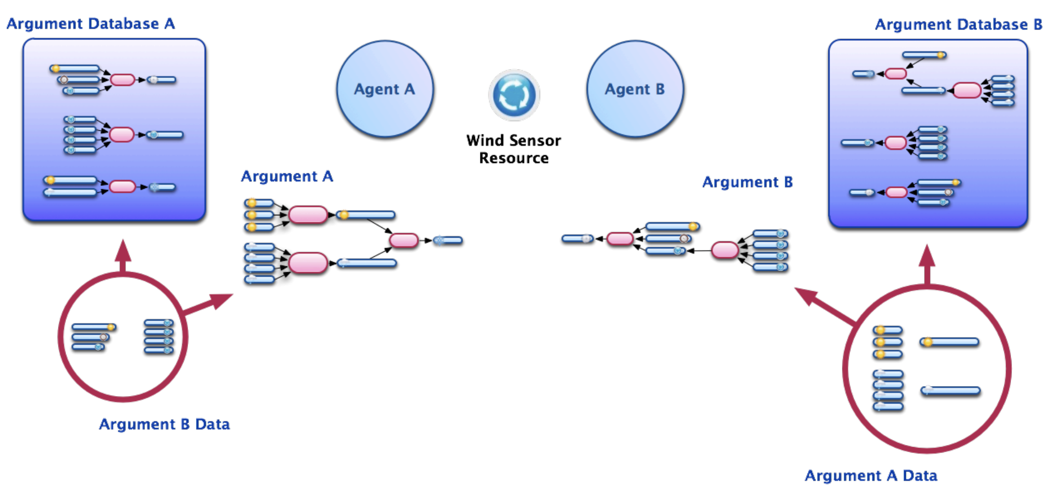

Figure 51. Each agent extracts the evidence used for its own argument and shares it with the other agent. The agents then update the arguments in their argument databases using the new evidence.

Another advantage of sharing evidence is that, upon its exchange, the agents would have all the information necessary to reach mutual agreement. While other negotiation protocols focus on competition, this protocol focuses on consensus and cooperation. The final expected result of such a negotiation is shown in Figure 52.

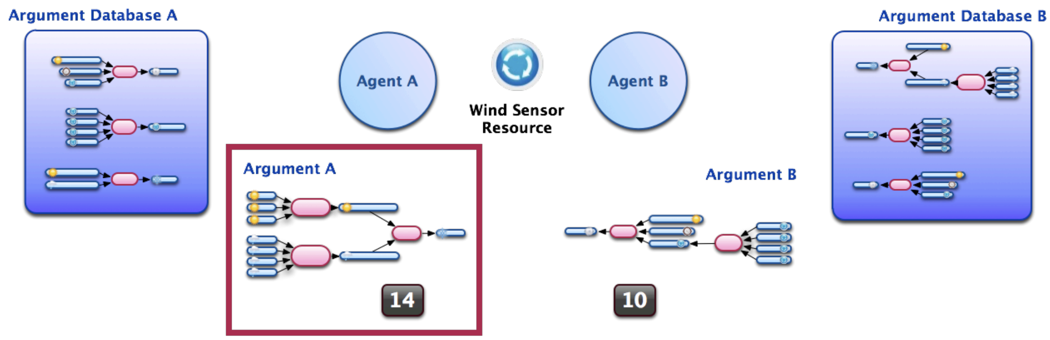

Figure 52. Following the exchange of evidence, the agents are able to reach consensus on the decision of which agent should receive the resource.

This future work is possible using this dissertation as a foundation for implementing arguments in EFA. Another important extension would be developing a database data structure to store agent argument collections. Agreement could still be determined using the SoD protocol. For example, if one agent agreed that the other agent needed the resource more urgently, it could simply acquiesce. A new protocol would be needed for the exchange and reevaluation step, should the agents disagree. Finally, a new concrete problem domain would need to be determined as a basis for experiment.

Bibliography

Amgoud, L., Parsons, S., & Maudet, N., 2000, Arguments, dialogue, and negotiation, Proceedings of ECAI-00.

Amgoud, L., Bodenstaff, L., Caminada, M., McBurney, P., Parsons, S., Prakken, H., van Veenen, J., & Vreeswijk, G., 2006, Final review and report on formal argumentation system. Deliverable D2.6, ASPIC IST-FP6-002307.

ArgMAS, 2009, Argumentation in Multi-Agent Systems. http://homepages.inf.ed.ac.uk/irahwan/argmas/argmas08/ (last accessed March 6, 2009).

Basir, O., & Yuan, X., 2007, Engine fault diagnosis based on multi-sensor information fusion using Dempster-Shafer evidence theory. Information Fusion, 8(4), 379-386.

Bex, F., Prakken, H., Reed, C., & Walton, D., 2003, Towards a Formal Account of Reasoning about Evidence: Argumentation Schemes and Generalisations. Artificial Intelligence and Law, 11(2-3), 125-165.

Black, T., 2008, Solving the Problem of Easy Knowledge. The Philosophical Quarterly, 58, 597-617.

Caminada, M., & Amgoud, L., 2007, On the evaluation of argumentation formalisms, Artificial Intelligence, 171, 286-310.

CMNA, 2009, Computational Models of Natural Argument. http://www.cmna.info/

CMNA8/ (last accessed March 6, 2009).

Cohen, J, 1988, Statistical Power Analysis for the Behavioral Sciences, Second Edition. Lawrence Erlbaum Associates.

COMMA, 2009, Conference on Computational Models of Argument. http://www.irit.fr/comma08/ (last accessed March 6, 2009).

Conee, E., & Feldman, R., 2004, Evidentialism: Essays in Epistemology. Clarendon Press.

Delin, K., Jackson, S., & Some, R, 1999, Sensor Webs. NASA Tech Briefs, 23, 80.

Delin, K., & Jackson, S., 2000, Sensor Web for In Situ Exploration of Gaseous Biosignatures. IEEE 2000 Aerospace Conference Proceedings 7 (18-25) 465-472.

Delin, K., & Jackson, S., 2001, The Sensor Web: A New Instrument Concept. SPIE Symposium on Integrated Optics. San Jose, CA 9.

Delin, K, 2002, The Sensor Web: A Macro-Instrument for Coordinated Sensing. Sensors, 2, 270-285.

Delin, K., Jackson, S., Johnson, D., Burleigh, S., Woodrow, R., McAuley, J., Dohm, J., Ip, F., Ferre, T., Rucker, D., & Baker, V., 2005, Environmental Studies with the Sensor Web: Principles and Practice. Sensors, 5, 103-117.

Dung, P. M., 1995, On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. Artificial Intelligence, 77, 321-357.

Dung, P. M., Kowalski, R. A., & Toni F., 2009, Assumption-based argumentation. In: Rahwan, I., Simari, G., editors. Argumentation in Artificial Intelligence. Springer.

Environment Canada, 2011, Public Alerting Criteria for Blizzard. Web site, http://www.ec.gc.ca/meteo-weather/default.asp?lang=En&n=D9553AB5-1#blizzard

Fox, J., 1981, Towards a reconciliation of fuzzy logic and standard logic. International Journal of Man-Machine Studies, 15, 213-220.

Fox, J., Glasspool, D., Patkar, V., Austin, M., Black, L., South, M., Robertson, D., & Vincent, C., 2010, Delivering clinical decision support services: There is nothing as practical as a good theory. Journal of Biomedical Informatics, 43(5), 831-843.

Gan, Q., & Harris, C., 2001, Comparison of two measurement fusion methods for Kalman-filter-based multisensor data fusion. IEEE Transactions on Aerospace and Electronic Systems.

Gordon, T., Prakken, H., and Walton, D., 2007, The Carneades model of argument and burden of proof. Artificial Intelligence, 171, 875-896.

Haack, S, 2009, Evidence and Inquiry: A Pragmatist Reconstruction of Epistemology. Cambridge.

Heavner, M., Fatland, D., EW. Hood, & Connor, C., 2008, A Cryospheric Sensor Web Use Case on a Small Temperate Glacier. Proceedings of ESTC-08.

Hong, X., Nugent, C., Mulvenna, M., McClean, S., Scotney, B., & Devlin, S., 2009, Evidential fusion of sensor data for activity recognition in smart homes. Pervasive and Mobile Computing, 5(3), 236-252.

Hurley, P., 2003, A Concise Introduction to Logic, Eighth Edition. Wadsworth.

Jade, 2011, Jade - Java Agent DEvelopment Framework. Web site, http://jade.tilab.com/

Jadex, 2011, Jadex BDI Agent System. Web site, http://jadex-agents.informatik.uni-hamburg.de/

Kedar, S., Chien, S., Webb, F., Tran, D., Doubleday, J., Davis, A., & Pieri, D, 2008, Optimized Autonomous Space In-situ Sensor-Web for Volcano Monitoring. Proceedings of ESTC-08.

Keppens, J. & Zeleznikow, J., (2003), A Model based Reasoning approach for generating plausible crime scenarios from evidence. Proceedings of the 9th International Conference of Artificial Intelligence and Law.

Kotrlik, J. W., & Williams, H. A., 2003, The incorporation of effect size in information technology, learning, and performance research. Information Technology, Learning, and Performance, 21(1), 1-7

Liu, X., Khudkhudia, E., & Leu, M., 2008, Incorporation of evidences into an intelligent Computational Argumentation network for a web-based collaborative engineering design system. International Symposium on Collaborative Technologies and Systems.

Madau, D.P., Yuan, F., Davis, L.I., Jr., Feldkamp, L.A., 1993, Fuzzy logic anti-lock brake system for a limited range coefficient of friction surface. Second IEEE International Conference on Fuzzy Systems, 2, 883 - 888.

McBurney, P., 2002, Games that agents play: A formal framework for dialogues between autonomous agents. Journal of Logic, Language and Information, 11.

Olfati-Saber, R., 2005, Distributed Kalman Filter with Embedded Consensus Filters. 44th IEEE Conference on Decision and Control.

Ontañón, S., & Plaza, E., 2010, Empirical Argumentation: Integrating Induction and Argumentation in MAS. Proceedings of ArgMAS-10.

Oren, N., Norman, T., & Preece, A., 2007, Subjective logic and arguing with evidence.

Oren, N., & Norman, T., 2008, Semantics for Evidence-Based Argumentation. Proceedings of COMMA-08.

Pavlin, G., Oude, P., Maris, M., & Hood, T., 2006, Distributed Perception Networks: An Architecture for Information Fusion Systems Based on Causal Probabilistic Models. IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems.

Pavlin, G., de Oude, P., Maris, M., Nunnink, J., & Hood, T., 2010, A Multi Agent Systems Approach to Distributed Bayesian Information Fusion. Information Fusion, 11(3) ,267-282.

Pollock, J., 1994, Justification and Defeat. Artificial Intelligence, 67, 377-408.

Poston, T., 2007, Foundational Evidentialism and the Problem of Scatter. Abstracta, 3(2), 89-106.

Prakken, H., & Sartor, G., 1997, Argument-based logic programming with defeasible priorities. Journal of Applied Non-classical Logics, 7, 25-75.

Prakken, H., 2010, An abstract framework for argumentation with structured arguments. Argument and Computation, 1, 93-124.

Rahwan, I., Zablith, F., & Reed, C., 2007, Laying the foundations for a World Wide Argument Web. Artificial Intelligence, 172, 10-15, 897-921.

Redford, C., & Agah, A., 2010, A Framework for Evidence-Based Argument Construction and Evaluation Applied to Multi-Agent Sensor Webs, International Conference on Artificial Intelligence and Pattern Recognition (AIPR-10), Orlando, FL

Redford, C., & Agah, A., 2011, Evidence-Based Argumentation versus Complete Sharing in Multi-Agent Sensor Webs, International Journal of Artificial Intelligence, 7, 47-62.

Redford, C., & Agah, A., 2012, Evidentialist Foundationalist Argumentation for Multi-Agent Sensor Fusion, Artificial Intelligence Review. Accepted. In Print.

Reed, C., & Rowe, G., 2001, Araucaria: Software for Puzzles in Argument Diagramming and XML. Department of Applied Computing, University of Dundee Technical Report.

Reed, C., & Wells, S., 2007, Dialogical Argument as an Interface to Complex Debates. IEEE Intelligent Systems, 22(6), 60-65.

Rosencrantz, M., Gordon, G., & Thrun, S., 2003, Decentralized Sensor Fusion with Distributed Particle Filters. Proceedings of UAI-03.

Scott, P., and Rogova, G., 2004, Crisis Management in a Data Fusion Synthetic Task Environment. 7th International Conference on Information Fusion.

Sherwood, R., & Chien, S., 2007, Sensor Web Technologies: A New Paradigm for Operations. Proceedings of RCSGSO-07.

So, A.T.P., Li, K.C., Chan, W.L., 1993, Implementation of fuzzy logic based automatic camera focuser. TENCON-93(4), 292-295.

Tsatsoulis, C., Ahmad, N., Komp, E., & Redford, C., 2008, Using a Contract Net to Dynamically Configure Sensor Webs. IEEE Aerospace Conference.

Vreeswijk, G., 1997, Abstract argumentation systems. Artificial Intelligence, 90, 225-279.

Walton, D., Reed, C., & Macagno, F., 2008, Argumentation Schemes. Cambridge University Press.

Wampler-Doty, M., 2010, Evidentialist Logic. MSc Thesis, Universiteit van Amsterdam, Netherlands. Retrieved from ILLC Publications (Accession Order No. MoL-2010-15).

Wolfram Mathematica, 2011, WeatherData function. Web site, http://reference.wolfram.com/mathematica/ref/WeatherData.html