A multi-agent trust framework for fusing subjective opinions with imperfect understanding in space domain awareness using the Scruff AI framework

Matthew P. Wilkins, Ph.D.
L3Harris Technologies, Inc.

Sanja Cvijic, Ph.D.
Charles River Analytics

Weston Faber, Ph.D.
L3Harris Technologies, Inc.

1. INTRODUCTION

It is widely acknowledged that the space domain is contested and congested. [1] Owing to high dollar outcomes and negative impact to critical missions, agents in the space domain rarely accept data from untrusted sources (i.e. commercial and non-traditional). To this end, methods to speed the timeline for integration of new sensor sources are being proposed based upon tiered approaches to trust but they do not present a general formalism to compute with trust. [2] In the meantime, agents continue to rely on "vetted" sensors to independently assess complex events and then make consequential decisions. "Vetted" sensors typically are those that agents have control over and have been calibrated to their own specifications. [3] Therefore, the agent's belief that correct decisions are being made is strictly shaped by the hard evidence (i.e. metric data plus uncertainty) provided by these sensors.

Unity of effort is 'the product of successful unified action' and consists of 'coordination and cooperation toward common objectives, even if the participants are not necessarily part of the same command or organization.' [4, 5] The barrier to cooperative decision making is the lack of a mechanism to incorporate soft evidence (i.e. subjective opinions) or perceived trust in the agent's decision-making process. Therefore, a framework is needed for operational decision-making that maps both hard and soft evidence to belief and certainty of action. Further, assessing the trustworthiness of something based upon the supplied evidence (i.e. another agent, a sensor source, individual data elements, etc.) is separate and distinct from the decision to trust, which is influenced by the agent's own experience (i.e., prior beliefs), knowledge (i.e. current evidence), and trust propensity (i.e., willingness to make yourself vulnerable). [6]

For the sake of discussion, let's examine a common problem in space domain awareness (SDA) that requires trust and cooperation. Our tracking systems have indicated that two objects are potentially on a collision path with a close approach distance below some keep-out threshold. We want to compute probability of collision between two objects. Each object could be tracked by one or more sensors together or separately. That is, there could be one track per object from the same sensor, one track per object from different sensors, or multiple tracks of each object from two or more sensors. In each of these cases, we can easily compute the probability of collision (Pc) using the uncertainty of the provided tracks. Yet even then, data from a trusted sensor could still have problems ranging from systemic errors intrinsic to hardware or even operator idiosyncrasies. In a collaborative environment, consider a situation where Country A tracks all space objects and identifies a conjunction between Country A and Country B assets. Country B supplies their own best track of their own object. Does Country A use their own track or the track provided by Country B to assess the collision risk and potentially maneuver out of the way? Does Country A trust the data that has been provide from an external source to make consequential decisions? Should Country A use the data in aggregate or separately to compare results?

The answer to these questions is twofold. First, one must evaluate trustworthiness of the source and/or the data itself. Second, one must actively make the decision to trust considering the trustworthiness assessment. Further complications often arise in SDA given that we don't usually know until after the raw data has been exploited to generate object tracks if there was an issue identified - particularly in a collaborative sensor setting. For example, there is a notion of "closely spaced objects" in the sensor field of view leading in some cases to cross-tagged or mis-tagged data association labels. For this scenario, we posit that trust in the data should be highest when the objects are clearly separable but worst when they are closest together. From one sensor's geometric perspective, it has a clear view of the two conjuncting objects but a different sensor may not. While each sensor may itself be assessed to be trustworthy, the data could be untrustworthy, or less valuable, based upon geometry for example. This scenario indicates we need a methodology to formulate and manage trust on an observation-by-observation basis not just a sensor-by-sensor basis.

Trust has been implemented as a computational framework in multiple fields – most notably in the field of wireless networks and cyber security where we are interested to know, for example, if a network node has been compromised with malicious code. [7, 8, 9] The space domain has yet to take advantage of recent advances in multi-agent trust frameworks where trust is no longer implicit but rather explicitly defined. For our purposes in SDA, we have discovered in review of the literature that there are several major theories that have been loosely coupled together through subject matter experts (SME) in a human-in-the-loop formulation. The goal of this paper is to provide an overview of the various techniques and theories in the literature that could be brought together to form and proposes an automated, rigorous, human-on-the-loop, end-to-end trust framework for SDA. This paper explores what is required to construct a formal trust framework where a decision-making agent must assess the trustworthiness of multiple information agents independently observing the same type of space domain event. We require that the decision-making agent be presented with potentially conflicting information from two or more information agents that have a variable history length of observations.

2. TRUST AND DECISION MAKING OVEREVIEW

In this section, we survey the current literature on the various theories required to achieve a fully realized trust framework. Trust formalisms provide a mechanism to compute with trust. However, we also need mechanisms to establish that data and sensors are trustworthy, which implies that we also require a model for quality and value of data. These trustworthiness models should be implemented in a way that allows us to flexibly reason with statistical rigor. Finally, we need a rigorous decision theory that connects the trust formalisms with our trustworthiness models and generates an explainable and prioritized set of courses of action in which a human decision maker has full confidence.

To begin, there are various methods to compute trustworthiness beginning with the seminal work of Marsh in 1994 who described a first formalism for trust. [10] A contemporary of Marsh, Jøsang went on to produce the first comprehensive treatment of subjective logic and its operations in 2016. [11] Of particular note is the work of Wang and Singh which provided a rigorous framework for assessing trustworthiness as the probability of a positive outcome following the early work of Jøsang. [12, 13] Additionally, Wang and Singh defined a probability certainty density function (PCDF) that assesses the strength of an agent's belief that the trustworthiness is a specific value. For the formalism to hold, it is required that certainty increases as evidence increases while certainty decreases as conflict between agents increases in the supplied evidence. For complex events involving multiple responsible parties with various subjective opinions and imperfect understanding, there are a variety of methods to fuse agent opinions and address conflicting opinions (i.e. soft evidence) as well as social distance of opinions (i.e. gossip) through discounting trustworthiness. [14, 15, 16]

To model trustworthiness, there is a significant body of work around the concepts of quality of information (QoI) and value of information (VoI) provided by sensor networks spearheaded by Bisdikian at IBM Corporation for the US Army Research Laboratory in 2009. The central idea is that data has quality that can be defined via ontologies independent of any situational context including not only uncertainty measures but also timeliness, completeness, and reliability measures. However, data only has value when it is assessed in context of its intended use as a function of its quality. Bisdikian et al. presented a first of its kind taxonomy categorizing QoI and VoI that were implemented using Unified Modeling Language (UML) constructs allowing for graphic representations of object-oriented data. [17, 18, 19] While QoI and VoI have been presented as a computational mechanism for things like sensor selection in tactical networks, [20] we assert that the underlying framework of UML, which seems to be the current standard in the literature, is ill suited for probabilistic and uncertain representations of the various taxonomy elements and their relationships to each other. An alternative to the UML framework found in the literature is probabilistic programming using Scruff, which will be discussed in detail later in the paper.

Lastly, a significant body of work has been established on the agent decision process as it pertains to trust. Once we have established the trustworthiness of something based upon evidence and assessed our certainty in this belief, we still must actively decide to trust by accepting the various risks and consequences of such a decision in context with the situation at hand. [10] Decision making has largely been the domain of the human operator of machine systems either in- or on-the-loop. As the complexity of systems, data, and consequences of decisions have grown, there has been a need to formulate the decision-making process in a rigorous algorithmic manner. A variety of methods broadly called Multi-Criteria Decision Making (MCDM) methods have been explored including Simple Additive Weightage (SAW), Analytic Hierarchy Process (AHP), Elimination Et Choice Translating Reality (ELECTRE), Preference Ranking organization Method for Enrichment Evaluations (PROMETHEE), and Technique for Order Preference and Similarity to Ideal Solution (TOPSIS), to name a few. Further there are "fuzzy" variations of many of these approaches that try to account for uncertainty in the decision-making process whether it be overall objectives or relative evaluation of importance criteria. [21, 22]

Out of all MCDM methods, the Analytic Hierarchy Process (AHP) developed by Saaty in 1980, is perhaps the most well studied and broadly utilized. AHP prioritizes alternatives (e.g., courses of action) based upon paired comparison of criteria relative to a goal. [23, 24] The pairwise assessments are placed into a square comparison matrix with 1's on the diagonal. For each off-diagonal element, $x_{ij} \forall\, j > i$, assign a value from 1 to 9, where 1 represents equal importance and 9 represents extreme favoritism of one over the other. Likewise, the corresponding element, x_{ji}, is the reciprocal value $x_{ji} = x_{ij}^{-1}$. Saaty was able to demonstrate that the principal eigenvector of such a comparison matrix is a vector of prioritization weights that forms the basis for a rigorous decision process. We will describe AHP in further detail below. While AHP has criticisms and detractors, AHP has been used in many real-world fields including government acquisition, marketing, resource allocation, policy making, and economics. [25] *Prima facie*, AHP is relatively straightforward to implement but typically requires a SME to properly assign the pair-wise comparison weights and AHP is typically used for single instance decision making with a single agent. [26] For decision making in near real time systems, AHP can be extended with a finite state machine and a scheduler model to facilitate dynamic, short-term decision making but it still does not accommodate multiple agents in the mix. [27]

With AHP as the technical backbone for decision making, various attempts to bring together these key concepts into a rigorous computational trust framework have been attempted over the last several decades. We initially focused on the work of Chan et al. who combined the QoI model of Bisdikian with the AHP process of Saaty and applied it to the tactical sensor source selection problem. [20] In our review of this and similar papers, as best as we can determine, there is no direct connection between the QoI hierarchy representation and AHP. Chan et al took the more detailed and complex QoI taxonomy from Bisdikian [19] and reshaped it into the layered relationship of criteria and results needed to conform to the basic AHP construct. There does not appear to be a direct way to connect an arbitrarily complex QoI and VoI taxonomy with AHP. Lastly, the output of AHP is a set of priority weights but guidance in the literature is thin on how to choose between similarly weighted options with unequal influencing criteria. Even if we were able to map a more complex QoI and VoI model into the AHP framework, it would still be a single instance decision mechanism, albeit one that may be useful for certain situations like evaluating on a granular level whether to act based upon multiple probability of collision calculations.

In order to extend trust to more complex scenarios where multiple agents are cooperating and sometimes competing to achieve a goal, we must turn to techniques like the Partially Observable Markov Decision Process (POMDP) which has only recently been explored in the context of trust-based decision making. A POMDP allows for a single agent to leverage their memory about previous actions and observations to determine current states and predict future rewards across various courses of action. A decision policy must be formulated that maps the agent's behavior as a function of the current state but solving for this policy is an undecidable infinite horizon problem that requires approximation techniques to determine a reasonable decision in near-real time. Expanding the POMDP to multiple agents, the decentralized POMDP (DEC-POMDP) allows for decision policies that depend upon all the policies and states of all the agents. [28] The Interactive POMDP (I-POMDP) extends the DEC-POMP framework to include the beliefs an agent has about other agents. [29] Fortunately for our purposes, Richard Seymour introduced the trust-based I-POMDP (TI-POMDP) in 2019, which is a novel approach to multi-agent cooperation that allows a group of agents to reason about the trustworthiness of each other and apply that trust to the decision policy. Seymour's testing showed a 3.8

times increase in average award for agents using the TI-POMDP decision model. [30] While the TI-POMPD closely aligns with our needs, one still must provide it with a functional trust model.

3. TRUST FRAMEWORK

Here we present a system view of a trust framework based upon the work of Govindan and Mohapatra on trust computations within mobile *ad hoc* networks (MANETs). [9] This paper will not address all of these services as they extend beyond the scope of our present work, but it is useful to understand all of the needed trust mechanisms that should be in place for a fully realized system. What we have done differently is to frame this as a modern microservice based architecture where both external and internal data streams first pass through a QoI and VoI annotator before being published on the data as a service (DaaS) common transport layer. Trust is established through various services, which will be briefly explained here.

1. QoI and VoI Services: Annotate incoming data streams with data quality assessments like bias determinations, uncertainty quantification, reliability, timeliness etc. Assess value in context with system mission intent defined by the user and microservices needing to consume the data products.
2. Trust Evaluator: Centralized or decentralized trust computations based upon metrics, definitions, and recommendations from inside and outside the system
3. Trust Propagator: Synchronize common operating picture (COP) for trust across system nodes
4. Trust Aggregator: Before and during trust propagation, combine trust assessments using weighted means and gossip aggregation across the system into a consolidated trust COP view that can be archived
5. Trust Recorder: Manage the Trust Store and consume trust data products for archival purposes
6. Trust Prediction: Predict situational trust given past history based upon state vector machines and/or Kalman filter based approaches
7. Security Services: In addition to trust informed data processing, this would be the most likely point of interface with the larger system where output products of the trust framework could include access control adjudication and nefarious/anomalous behavior detection

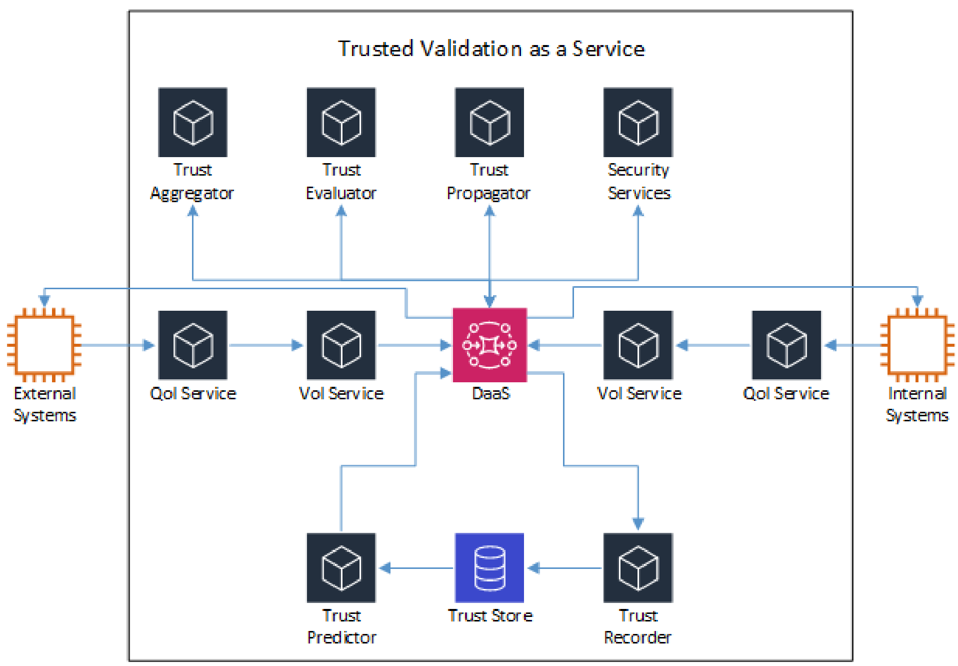


Figure 1: Notional Trust Framework

The QoI and VoI microservices depicted in Figure 1 are not necessarily serial microservices but rather are drawn in order of their annotation precedence. Figure 2 depicts the interaction between the QoI and VoI microservices. All data entering the Data-as-a-Service layer, whether from internal or external sources, will have some sort of quality metrics that can be independently assessed and annotated. From there, data products can have value assessments levied in context with specific uses coming from end users or other applications and microservices within the trust framework.

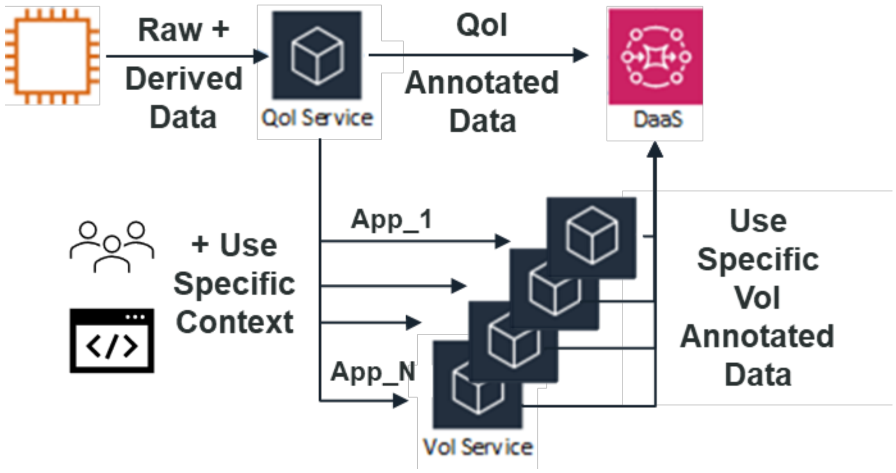


Figure 2: Interaction between QoI and VoI microservices

4. TRUST MODELING

When we say that we "trust" someone or something, we are combining both our assessment of that thing and our decision to trust all in one turn of phrase. At this point it is worth reinforcing some key definitions as it is common to use the word "trust" in a variety of contexts as both a noun and a verb. [31] The act of trust is a decision that must be made to become vulnerable to someone (i.e. taking a risk) even when they are already deemed to be trustworthy. We will cover the decision to trust in a later section on the analytic hierarchy process. In this section, we are interested in determining trustworthiness, which are the characteristics and behaviors of an agent's honesty, fairness, and/or benevolence that inspires positive expectations by another agent. For our purposes, trustworthiness is the belief that another agent will behave as expected, which includes things like fulfilling a contract governed by an interface control document (ICD) in a timely and reliable fashion. We are focused on evidence-based approaches that allow for multi-agent systems to accept trust reports from multiple sources. From a systems architecture perspective, it is conceivable that a reputation agency could be developed to manage trust on behalf of all the agents. The reputation agency would be responsible for sharing the internal opinions of agents with each other so that they can potentially adjust their own opinion accordingly. This is the basis of rating systems like Yelp and eBay where potential customers can see the ratings of established customers to inform their decision to purchase. However, at this level of formalism, we are focused on direct agent to agent interactions so that we can minimize any confusion surrounding referred evidence or "gossip." This is what we will measure as the probability of a positive outcome given a situation and a set of evidence represented by a history of positive and negative experiences with a given agent.

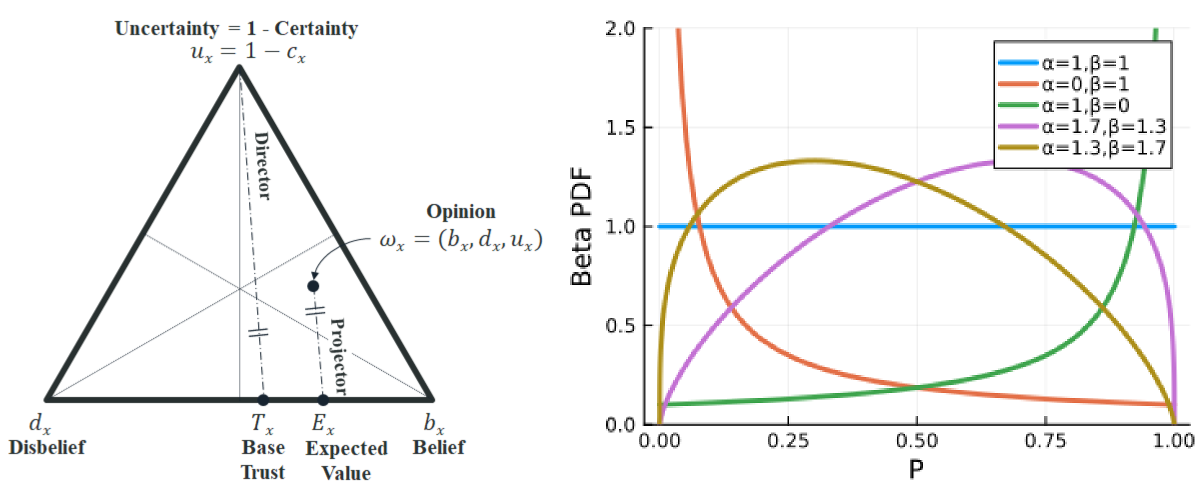
Table 1 presents the basic trust notation and equations that we will employ following [10, 11, 32]. We assume for the sake of notation a set of events or situations, $\lambda_i \; \forall \; i > 0$, along with a set of agents, $A = \{x, y, z, \dots\}$, that will have their own trust assessment of each situation as well as each other. Each situation will have some level of importance, $I_x(\lambda_i)$, to the agent that will influence decision making later in the paper. We will generally leave off the situational notation as long as it's clear that we are referring to a singular event or situation throughout. Each agent, noted with a subscript, will have a basic trust level, T_x, that represents its trust propensity as a function of the agent's entire life experiences. A value of 0.5 would represent no trust (i.e. no opinion one way or the other) while 1 is complete trust and 0 is complete distrust. The higher an agent's basic trust level, the more likely they will trust in the absence of other evidence. Here you would find terms like pessimism ($T_x < 0.5$), optimism ($T_x > 0.5$), and realism ($T_x = 0.5$) appropriate to reflect how much and how often this basic trust level changes with evidence. For our purposes, we will assume that base trust is a uniform uninformed prior of 0.5. Updates to base trust to reflect optimism and pessimism are beyond the scope of this paper but we refer you to the literature for details on how to evaluate trust deltas as a function of world view.

For the sake of discussion, assume we have a situation λ_i representing a potential collision between two satellites and this will occur with probability $p \in [0,1]$. We know from the Space Flight Safety Handbook for Satellite Operators how the United States Space Command (USSPACECOM) computes the probability of collision between two conjuncting satellites. [33] However, this is reliant upon specific methodologies and best practices along with the Space Surveillance Network (SSN) data, which is generally not available to anyone outside the US military. Other space actors might use different methods and have different data to compute a probability of collision about the same situation. We will call USSPACECOM as agent x. Consider the situation where an external observer, agent y, provides agent x with their opinion regarding the potential conjunction. A subjective binomial opinion regarding the truth of this situation is tuple comprised of belief that the situation is true, disbelief in the situation, and the uncertainty, $\omega_x(\lambda_i) = (b_x, d_x, u_x)$. We can say that the agent x's trust has a trust space in variable p regarding situation λ_i, which is modeled as a three-dimensional space of reals in [0, 1] represented by weights assigned to belief, disbelief, and uncertainty (i.e. 1 – certainty) where unity and zero in trust space represent perfect knowledge and ignorance, respectively.

$$T_x(\lambda_i) = \{(b_x, d_x, u_x) \quad | \quad 0 < b_x, d_x, u_x < 1, \qquad b_x + d_x + u_x = 1\}$$

Binomial opinions are graphically represented as a triangle with belief, disbelief, and uncertainty on each vertex as shown in Figure 3a. A base rate equivalent to an agents' basic trust level, T_x, is shown on the base line with a director line pointing at the uncertainty vertex. In some parts of the literature, an opinion is expressed as the opinion tuple plus the base rate, $\omega_x(\lambda_i) = (b_x, d_x, u_x, T_x)$. For this paper, we will stick with the core opinion tuple. For any opinion, we can find the expected belief by projecting it onto the baseline in the direction of the director line using the definition $E_x = b_x + T_x u_x$. When uncertainty is zero, we say that the belief is dogmatic. When the belief or disbelief is unity, we have the equivalent to a binary logical TRUE or FALSE, respectively.

A probability density function (pdf) of the probability of a positive experience (i.e. the agent's belief regarding the situation is true), $f(p)$, is defined such that $\int_0^1 f(p)dp = 1$. This probability density function can be conditionally updated given the agent's table of situational evidence with our initial pdf being the uninformed uniform prior without evidence. From Subjective Logic, [11] this binomial opinion has a distribution equivalent to the Beta or Dirichlet, as shown in Figure 3b, which is defined on the range [0,1]. Notice that Beta(1,1), implying balance in positive and negative evidence, corresponds to the uniform distribution on [0,1] and zero elsewhere. The agent's current trust level, therefore, corresponds to increasing deviation from the uninformed prior (i.e. uniform) distribution. An agent's trust is also a function of how strongly the agent believes a positive experience will occur. Following [11], this is called a probability certainty density function (PCDF) given by $c_f = \frac{1}{2}\int_0^1 |f(p) - 1| dp$, which represents the deviation of the agent's trust from the mean absolute deviation.



(a) (b)

Figure 3: Pictorial representation of trust metrics. (a) Opinion tuple (belief, disbelief, and uncertainty) represented as a triangle (b) Beta distribution with various choices of positive and negative parameters (α,β)

Table 1: Basic Trust Notation and Equations [10, 11, 31]

	Description	Representation	Value Range
1	Set of Situations	$\Lambda = \{\lambda_i \; \forall \; i > 0\}$	
2	Set of Agents	$A = \{x, y, z, \dots\}$	
3	Importance (e.g. of λ_i to x)	$I_x(\lambda_i)$	[0, 1]
4	Basic Trust (e.g. of x w/o evidence)	T_x	[0, 1]
5	General Trust (e.g. of x in y)	$T_x(y)$	[0, 1]
6	Evidence (positive, negative)	$(r, s), \mathrm{t} = \mathrm{r} + \mathrm{s}$	$r, s > 0$
7	Beta PDF	$f(p, \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}$	[0,1] →[0,∞)
8	Probability of a Positive Outcome for x given the Evidence (r,s)	$\alpha = r + 2T_x, \beta = s + 2(1 - T_x)$ $f(p \vert (r, s)) = \frac{\Gamma(r + s + 2)}{\Gamma(r + 2T_x)\Gamma(s + 2(1 - T_x))} p^{r+2T_x-1}(1-p)^{s+2(1-T_x)-1}$ $= \frac{p^{r+2T_x-1}(1-p)^{s+2(1-T_x)-1}}{\int_0^1 p^{r+2T_x-1}(1-p)^{s+2(1-T_x)-1} dp}$	[0,1] →[0,∞)
9	Certainty (1 – Uncertainty)	$c_x = 1 - u_x = \frac{1}{2}\int_0^1 \lvert f(p) - 1 \rvert dp$ $c_x(r, s) = \frac{1}{2}\int_0^1 \left\lvert \frac{p^r(1-p)^s}{\int_0^1 p^r(1-p)^s dp} - 1 \right\rvert dp$ $c_x(t) = c_x((t + 2)E_x - 1, (t + 2)(E_x - 1) - 1)$	[0,1]
10	Expected Probability of a Positive Outcome (Situational Trust)	$E_x(f(p, \alpha, \beta)) = \frac{\alpha}{\alpha + \beta}$ $E_x(\lambda_i \vert (r, s)) = \frac{r + 2T_x}{r + s + 2}$ $E_x(\lambda_i \vert (b, d, u)) = T_x(y, \lambda_i) = b + T_x u$	[0,1]
11	Opinion Tuple (e.g. of x for λ_i)	$\omega_x(\lambda_i) = (b_x, d_x, u_x)$	$b, d, u > 0$ $b + d + u$ $= 1$
12	Opinion Tuple Given Evidence	$\omega_x(\lambda_i \vert (r, s)) = (b_x(r, s), d_x(r, s), u_x(r, s))$ $= (E_x c(r, s), (1 - E_x)c(r, s), 1 - c(r, s))$	$b, d, u > 0$ $b + d + u$ $= 1$
13	Discounted Opinion (e.g. of x in y for λ_i because of $T_x(y)$)	$\omega_{x:y}(\lambda_i) = \omega_y(\lambda_i) \otimes T_x(y)$ $= (b_y T_x(y), d_y T_x(y),$ $1 - T_x(y) - u_y T_x(y))$	$b, d, u > 0$ $b + d + u$ $= 1$
14	Evidence Given an Opinion	$E = \frac{b}{b + d}, t_1 = 0, t_2 = r_{max} + s_{max} = t_{max}$ $while\ (t_2 - t_1) \geq \epsilon\ \{$ $t = \frac{t_2 + t_1}{2}$ $if\ c(t) < 1 - u:\ t_1 = t\ else\ t_2 = t\ \}$ $r = ((t + 2)E - 1), s = t - r$	$r, s > 0$
15	Opinion Fusion	$\oplus\,(\omega_x, \omega_y, \omega_z, \dots) = \omega(\lambda_i \vert \left(\sum_{i=0}^{n} r_i, \sum_{i=0}^{n} s_i\right))$	$b, d, u > 0$ $b + d + u$ $= 1$

Each agent will also have a trust level, $T_x(y)$, for each of the other agents representing the history of their pair-wise interactions. If agent y provides an opinion to agent x on the potential collision, we can write trust as

$$T_x(y,\lambda_i) = \{(b_x(y), d_x(y), u_x(y)) \quad | \quad 0 < b_x(y), d_x(y), u_x(y) < 1, \quad b_x(y) + d_x(y) + u_x(y) = 1\}$$

However, we may not want to directly use this opinion because our opinion of the reliability of agent y needs to be taken into account. We do this via a discounting operator $\otimes$ for agent x's trust in agent y, $T_x(y)$, as follows. [11]

$$\omega_{x:y}(\lambda_i) = (b_x(y), d_x(y), u_x(y)) = \omega_y(\lambda_i) \otimes T_x(y)$$
$$= (b_y T_x(y), d_y T_x(y), 1 - T_x(y) - u_y T_x(y))$$

For the satellite conjunction situation and based upon our current SDA practices, we use a two or three dimensional probability of collision theory as appropriate and assign that value as follows: $b_x = p_c, d_x = 1 - p_c, u_x = 0$. Our trust then is a dogmatic belief that the probability of collision is a specific value, p_c. The SDA community has traditionally not computed a belief uncertainty to go along with our probability of collision computations. Typically, the SDA community turns directly to risk assessment concepts to combine the probability of collision with some human centric assessment of collision consequence to determine if we want to act. [34] It is important to note that while we leverage object track uncertainty in the calculation of p_c, metric uncertainty is not the same thing as belief uncertainty. In a trust framework, our belief uncertainty should encapsulate positive and negative experiences related to the ultimate outcome of potential conjunctions. For example, if we believed a collision would not occur leading to a decision to not maneuver and subsequently there was no collision would be tallied as a positive experience. If a collision did occur in contravention to what our p_c metric indicated, that would be a negative experience.

Now consider an evidence space, E_x, is modeled for convenience as a two-dimensional space of reals corresponding to the number of positive (*r*) and negative (*s*) outcomes where these outcomes are based upon agent x's table of evidence.

$$E_x = \{(r,s) \quad | \quad r,s \geq 0, \quad t = r + s > 0\}$$

Generally, our trust evidence would be supplied via the trust computations described in the system framework above. For example, one could produce a trust report populated with metadata on data quality interactions like compliance with ICDs, timeliness, and responsiveness in a C2 context, for example. From our review of the literature, there is an open question of how to map a complex mixture of metadata representations into a simplistic positive or negative tally. For this paper, we will assume that these reports can be converted into equally weighted positive and negative interactions through our QoI model presented below and that the positive and negative reports can then be counted as *r* and *s*, respectively, with *r* and *s* typically represented as integers. We will discuss in the next section how to map QoI and VoI into evidence and certainty of belief.

Now we must map trust to evidence, and vice versa, with respect to opinions. The transformation between Evidence and Trust spaces is a bijection given by the following:

$$Z_x^\alpha(r_x(y), s_x(y)) = (b_x(y), d_x(y), u_x(y)) \therefore Z_x^\alpha(y) = \big(b(r,s), d(r,s), u(r,s)\big)$$

Marsh and Jøsang present a simplified closed form solution for Z^{-1} as $(b,d,u) = \left(\frac{r}{t+1}, \frac{s}{t+1}, \frac{1}{t+1}\right)$. However, this approach assumes constant certainty, $c = \frac{t}{t+1}$. The approach of Wang and Singh allows for computed uncertainty based upon the evidence for which there is no closed form solution. They provide an iterative algorithm which is shown in Table 1. [12] Agent trust in the probability of a positive outcome can then be represented by the following:

$$b(r,s) = f(p|\langle r,s \rangle) = \frac{p^{r+2T_x-1}(1-p)^{s+2(1-T_x)-1}}{\int_0^1 p^{r+2T_x-1}(1-p)^{s+2(1-T_x)-1} dp} = E_x c(r,s)$$

$$u(r,s) = 1 - c(r,s) = 1 - \frac{1}{2}\int_0^1 \left| \frac{p^{r+2T_x-1}(1-p)^{s+2(1-T_x)-1}}{\int_0^1 p^{r+2T_x-1}(1-p)^{s+2(1-T_x)-1} dp} - 1 \right| dp$$

$$d(r,s) = 1 - b(r,s) - u(r,s) = (1 - E_x)c(r,s)$$

The mapping from opinion evidence to the beta distribution is $\alpha = r + 2T_x, \beta = s + 2(1 - T_x)$. Having defined a mechanism to go from evidence reports to trust distributions, once the posterior trust distribution has been generated, it may be desired to notify the agent of any "significant" change in trust based upon the supplied evidence. The question becomes how to quantitatively decide if the change between the prior and posterior trust distributions is statistically meaningful. If an agent decides to create a change detection alert, how do they assess whether that alert is within a desired false alarm percentile? Fortunately, the authors have previously developed a truncated Sequential Probability Ratio Testing (TSPRT) method that relates information divergence to Type I and Type II errors. [35] The TSPRT method allows one to set a threshold for change detection based upon rigorous false alarm metrics.

5. MODELING QUALITY AND VALUE OF INFORMATION

Whether you are assessing the trustworthiness of someone or something or are making a decision to trust, we have a fundamental need to understand the quality and value of the data that we are basing our analysis upon. Even in situations of contractual trust where you have no choice but to use the data you are given, we can still monitor the quality of data over time to check for biases and inconsistencies that might influence results. For SDA purposes, we require automated systems for assessing sensor and data quality and value owing to large volumes of data produced by the SSN and supporting commercial data providers. Further, we need to be able to directly leverage the models for quality and value in our larger trust framework.

Multiple definitions for data quality have been proposed [36] but it was only recently that it was recognized the models for data quality and value needed to be clearly delineated. Following the work of Bisdikian et al. along with Chan et al. [17, 18, 19, 20], we present one possible hierarchical model for data quality and another for data value in Figure 4. Because these models will necessarily change over time, we need a flexible mechanism to represent and reason with them. Ruttenberg, Wilkins and Pfeffer demonstrated the use of the open-source Figaro probabilistic programming language (PPL), developed by Charles River Analytics, as applied to reasoning about resident space object (RSO) characterization based upon hard and soft evidence. [37] PPLs provide native probabilistic constructs and methods that encode the model complexity in the language, facilitating a more natural model building procedure. Because the probabilistic constructs are interwoven into the language, arbitrary data structures can be incorporated into models, and the object–oriented nature of some PPLs provides a natural means to express hierarchical models.

While the use of the Figaro PPL as applied to hierarchical reasoning on RSOs was innovative in prior work, there were and still are representational challenges that had to be overcome to enable flexible and modular probabilistic hierarchies. Since that time, Charles River Analytics has made strides in their ability to compute with hierarchical models with the advent of Scruff. Fortuitously, Scruff not only can more easily represent and reason with hierarchical models, it also is squarely aimed at modeling of agents empowered by artificial intelligence (AI). The following is a brief overview of Scruff's capabilities and its value for this purpose should be evident.

Here were present Scruff, also developed by Charles River Analytics, which is an AI framework to build agents that sense, reason, and learn in the world using a variety of models. Scruff, in many respects a second-generation PPL that learned from the limitations of Figaro. Scruff aims to integrate many kinds of models in a coherent framework, provide flexibility in spatiotemporal modeling, and provide tools to compose, share, and reuse models and model components. Scruff is provided as a Julia package and is licensed under the BSD-3-Clause License. The value of the proposed framework to SDA will not only enable an agent to make more trustworthy decisions in the presence of conflicting information but also enable cooperation and coordination based on statistical measures.

Scruff is a flexible framework for building AI systems to build agents that sense, reason, and learn in the world using a variety of models. [38] Although its roots are in probabilistic programming, it is not strictly speaking a probabilistic programming language. Instead, it is a framework for combining models of different kinds and reasoning with them. The name Scruff derives from the old debates in AI between the neats and the scruffies. Neats believed that unless systems were developed in a coherent framework, it would be impossible to scale development of AI systems to complex real-world problems. Scruffies believed that real-world problems require a variety of techniques that must be combined as best as possible, and forcing everything into a neat framework would hinder progress. We believe that

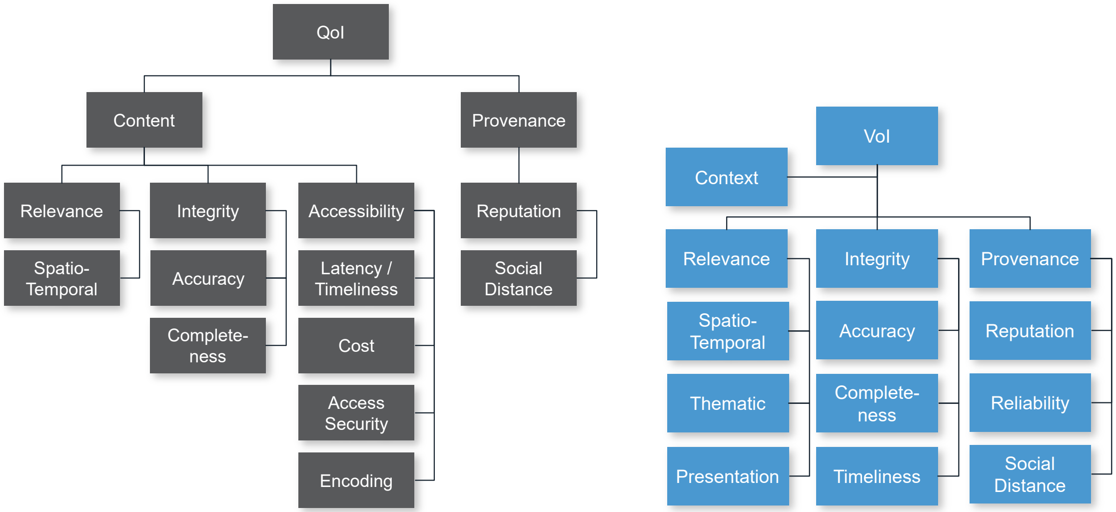


Figure 4: Example QoI and VoI Hierarchies

both camps have an element of the truth, and Scruff is an attempt to provide the best of both worlds. Scruff's philosophy is to allow a variety of representations and implementations to coexist side by side, and not every algorithm can be applied to every representation. However, they all coexist in a clean, well-defined and organized framework that enables scalable development of models and systems. Scruff is provided as a Julia package and is licensed under the BSD-3-Clause License.

Scruff provides three main features:

1. The ability to combine different kinds of models and reason with them using an algorithm in an integrated way. Scruff decomposes the representation of models from algorithms that work with them using operators. Any representation (the scruff word is sfunc (stochastic function, pronounced "essfunk")) that implements the operators can appear in algorithms. Using this approach enables us to generalize algorithms like belief propagation and importance sampling that have traditionally been applied to probabilistic models. A given sfunc does not have to support all operators and algorithms can use sfuncs in the appropriate way. For example, it is legal to have an sfunc that you can't sample from, which would not be possible in a typical probabilistic programming language.

2. A flexible framework for inference using these representations. Scruff distinguishes between the notion of a variable, which represents a value that can vary over time, and an instance of that variable, which represents its value at a particular time. In Scruff, variables are associated with models, which determine which sfunc to use for specific instances. There is no requirement that instances follow a regular time pattern; if the model supports it, instances can appear at any time interval. It is also possible to combine instances appearing at different time intervals, for example slowly changing and rapidly changing variables. Scruff also provides the ability to perform iterative inference, where beliefs about instances are refined through repeated computation.

3. Composition, reuse, and experimentation with different models, sfuncs, and algorithms. Scruff comes with an extensible and structured library of models, sfuncs, operators, and algorithms, making it easy to mix and match or extend with your own. For example, it is possible to implement alternative versions of an operators for an sfunc side by side and choose between them manually, or even automatically based on the characteristics of the specific instance. Another example is to compare accuracy and runtime between different time granularities on a variable by variable basis. Finally, as sfunc composition is highly structured, it is possible to experiment with specific sfunc choices in a systematic way.

The central concepts of Scruff are:

- Sfuncs, or stochastic functions, which represent mathematical relationships between variables (e.g., *Cat* for categorical, *Normal* for Gaussian, *DiscreteCPT* for discrete conditional, etc.)
- Operators, which define and implement computations on sfuncs (e.g., *sample*, *pdf*, *cpdf*, etc)
- Models, which specify how to create sfuncs in different situations (e.g., *static* for static behaviors, *homogenous* for behaviors that update at the constant time granularity, *time variable* for behaviors that change asynchronously)
- Variables, which represent domain entities that may take on different values at different times
- Networks, which consist of variables and the dependencies between them
- Instances, which represent a specific instantiation of a variable at a point in time
- Algorithms, which use operations to perform computations on networks (e.g., *particle filter*, *belief propagation*)
- Runtimes, which manage instances as well as information used by algorithms

6. DECIDING TO TRUST

We present two theories in detail for trust-based decision making where our CoAs could be as simple as trust vs do not trust or more complicated choices where various actions are delineated as a function of trust in the data or not. Out of the many MCDM theories in the literature, AHP stands out for multiple reasons. First, AHP is widely studied and applied across many fields. Second, AHP is structured in a hierarchical manner (hence its name), which is conducive to our QoI and VoI modeling. Third, AHP has straightforward intuitive solutions that are easy to comprehend which aids in human on-the-loop confidence in the system. Fourth, AHP easily accommodates both metric and semantic criteria by conversion to ratio scales. [25] As mentioned above, while AHP has its detractors, it is still the go-to decision tool for single agent decision making like the collision avoidance problem in SDA. In that instance, we have a variety of courses of action to choose from that are influenced to various extent by criteria that can be expressly enumerated. We only need to move to a more complex decision tool like the TI-POMDP when we want to allow for multiple agents working cooperatively or for the case of near-real time decision making in-line with raw data processing like the case of closely spaced objects in the field of view. Both techniques appear to have their place in a holistic trust framework.

6.1 Analytic Hierarchy Process (AHP)

In AHP, all problems are structured as shown in Figure 5. For any given objective, we establish the set of criteria, C_i, that affect/define the overall goal. We assume that this is an exhaustive list. Then, for each course of action (CoA) to achieve the goal, a human subject matter expert (SME) must assign a comparative judgement, a_{ij}, of the relative importance of one criteria, C_i, over another, C_j, according to the scale provided in Table 2. By definition, $a_{ji} = a_{ij}^{-1}$ and $a_{ii} = 1$. Saaty established the weights of the criteria can be calculated by solving for the principal eigenvector and associated eigenvalue according to

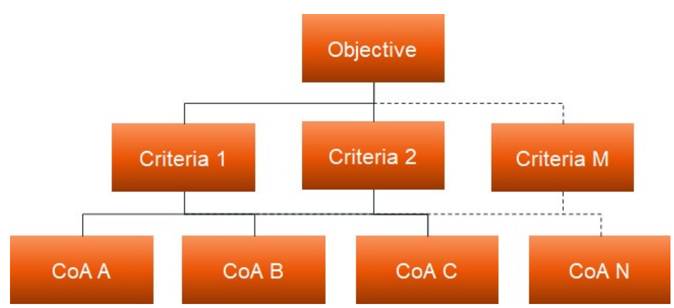


Figure 5: AHP Decision Hierarchy

$$AW = \lambda_{max} w$$

where A is the matrix of comparative judgements and λ_{max} is the largest eigenvalue of A associated with the real valued positive eigenvector w. When the eigenvector, w, is normalized, $\hat{w} = {}^{w}/_{\|w\|}$, it is the AHP vector of priorities with respect to the goal. Following this procedure for each CoA, generate the weights of alternatives for each criteria.

Note that the SME must make $\frac{M(M+1)}{2}$ judgements for each of the CoAs for a total of $\frac{M(M+1)N}{2}$ judgements. One of the criticisms of AHP is that this can be fatiguing and unreliable for a human to make this many assessments. We can now compute the weighted ranking for each of the CoAs using a weighted sum of the weight of the CoA_i with respect to C_j multiplied by the weight of C_j with respect to the objective.

$$w_{CoA_i} = \sum_{j=1}^{M} \widehat{w}_{CoA_i:C_j} \times \widehat{w}_{C_j:Objective} \; \forall \; i = 1, \ldots, N$$

Another criticism of AHP is that the relative comparison process can be inconsistent because of some redundancy in the process. For this, Saaty implemented a consistency check to determine if there is a reason to go back and double check the SME's comparative assessments. For each and every A matrix, we calculate a consistency ratio, CR. High inconsistency indicates a lack of information or lack of understanding on the part of the SME. Generally, we want CR < 0.1 where RI is the random consistency index that is the average of 500 randomly filled in matrices provided in Table 3. [23, 24]

$$CR = \frac{\lambda_{max} - M}{(M-1)RI} < 0.10$$

Using the AHP process above generates a ranked set of CoAs but the default choice is the highest rank CoA. This seems unsatisfying for our purposes in SDA particularly when the CoAs could have similar weightings but potentially unequal consequences. Going back to the seminal work of Marsh, he proposed that the decision to cooperate between agents can be thought of as a situational threshold that is a function of perceived risk, perceived competence among agents, our propensity to trust, and the trust one agent places in another. We propose that this same decision threshold could be applied to the AHP ranked choice problem. Marsh defines a cooperation threshold between agents as follows [10]

$$T_x(y, \lambda_i) > Cooperation\_Threshold_x(\lambda_i) \rightarrow Will\_Cooperate(x, y, \lambda_i)$$

$$Cooperation\_Threshold_x(\lambda_i) = \frac{Perceived\_Risk_x(\lambda_i)}{Perceived\_Competence_x(y, \lambda_i) + \widehat{T_x(y)}} \times I_x(\lambda_i)$$

Table 2: AHP Comparison Scale [22]

Intensity of Importance	Definition	Explanation
1	Equal Importance	Two activities contribute equally to the objective
2	Weak or Slight	
3	Moderate Importance	Experience and judgement slightly favor one over another
4	Moderate Plus	
5	Strong Importance	Experience and judgement strongly favor one over another
6	Strong Plus	
7	Very Strong or Demonstrated Importance	An activity is favored very strongly over another, and its dominance has been demonstrated in practice
8	Very, Very Strong	
9	Extreme Importance	The evidence favoring one over another is of the highest possible affirmation

Table 3: Random Consistency Index [23]

Random Consistency Index (RI)										
M	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.9	1.12	11.24	1.32	1.41	1.45	1.49

Where $I_x(\lambda_i) \in [0, +1]$ represents the importance of situation (λ_i) as viewed by Agent x, and $\widehat{T_x(y)}$ is the mean trust that agent x places in agent y across any known set of situations. The basic idea behind the formulation is that the more important a situation is to the agent, the trust level required to cooperate will necessarily be higher. Likewise, an agent's perceived risk is a subjective measure of the situational consequences of getting it wrong. We would expect that as the situational risk increases, that the trust threshold required to cooperate would increase. Risk is counterbalanced by the perceived competence of the other agent and the mean trust in that agent. While we can compute the basic trust based upon the agent's table of evidence, we will need a model for the competence. For our purposes in SDA as applied to AHP, we propose to compute the perceived competence factor with our QoI model and the importance factor with VoI.

$$Decision\_Threshold_x(\lambda_i) = \frac{Perceived\_Risk_x(\lambda_i)}{Perceived\_Competence_x(QoI, \lambda_i) + \widehat{T_x(y)}} \times I_x(VoI, \lambda_i)$$

When assessing perceived risk, one must consider the agent's experience with situation λ_i. If the agent has no knowledge or experience with the situation, we can assign risk based upon the agent's trust in itself during situational of ignorance, $Perceived_{Risk_x}(\lambda_i) = f(T_x(x, \lambda_i))$, which can then be updated after each decision outcome. On the other end of the experience spectrum, if the agent has considerable knowledge of λ_i, we can enumerate known risks and use a Bayesian theory to arrive at a useable metric (i.e. use the expected probability of each outcome). [10] Following this threshold based decision process, we can establish a rigorous metric to choose between CoAs as well as establish a minimum criteria for action whereby the system knows to wait for more data of the right quality and value before proceeding.

6.2 Trust Based Interactive Partially Observable Markov Decision Process (TI-POMDP)

An alternative approach would be to formalize the problem in terms of Markov Decision Processes (MDPs). Groundwork by Seymour [30] provides an advanced framework for modeling decision-making in multi-agent environments where trust and cooperation are key factors. This framework called Trust-based Interactive Partially Observable Markov Decision Process (TI-POMDP) builds upon MDPs and Partially Observable MDPs (POMDPs) to incorporate trust. In an MDP, an agent makes decisions in a stochastic environment, aiming to optimize the decisions according to an underlying objective. MDPs are characterized by a tuple consisting of states, actions, transition probabilities, and rewards. However, MDPs assume that the agent has full observability of the current state, which is not always realistic. POMDPs extend MDPs to situations where the agent can not directly observe the state of the environment in its entirety. Agents instead receive observations, often noisy, that provide partial information about the state. The agent must then maintain a belief state—a probability distribution over possible states—to make informed decisions. The TI-POMDP further extends the POMDP framework by incorporating the dimension of trust into multi-agent interactions. In cooperative systems, agents work towards a common goal but may face the risk of other agents being faulty or intentionally deceitful. The TI-POMDP allows agents to reason about the trustworthiness of their peers and decide whether to cooperate based on trust levels.

The TI-POMDP is defined by a tuple of six elements $\langle IS_i, A, T_i, \Omega_i, R_i \rangle$ where,

- IS_i is the set of interactive states $S \times M_j$, with S being the set of environment states, and M_j the set of models of agent j. This is an exhaustive representation of the underlying state space and an overlaying set of beliefs consisting of the different models the agents have for one another.
- A is the joint set of actions from all agents
- T_i is the transition model
- Ω_i is the set of all possible observations that agent i can make
- O_i is the probability that agent i makes an observation predicated on its state and actions
- R_i is the reward function

For example, let's compose this tuple for a two agent SDA conjunction scenario to show how the TI-POMDP framework can aid in providing a rigorous approach to modeling complex decision-making scenarios. In this scenario, two agents have differing beliefs regarding the states of the objects in conjunction leading to differences in subsequent probability of collision P_c calculations. To incorporate trust let's assume that the agents have a predetermined set of guidelines that define who will maneuver in instances of conjunction. Agents need to decide either to adhere to guidelines or to deviate from them to minimize P_c. In this scenario,

- $S = R^n$ where $n = 6$, for each state dimension, specifying the underlying possible positions and velocities of the Resident Space Objects (RSOs). Each agent maintains two "catalogs" – multi-target pdfs – one representing their understanding of the environment and another representing their understanding of the other agent's belief
- A is the joint set of actions consisting of each agent adhering to guidelines or deviating from guidelines.
- T_i each agents transition model consisting of non-linear orbital dynamics to include potential maneuver models according to the predefined guidelines
- $\Omega_i = R^m$ where m is the dimension of the measurement space
- O_i is the probability of the measurements occurring given the underlying state beliefs and transition models. These observations, although uncertain, allow agents to infer whether other agents are following guidelines thus enabling trust models to be updated overtime
- R_i is then $\min_{A} P_c$ and can be tailored to incorporate VoI and QoI allowing for those values to maximized while minimizing P_c

7. CONCLUSIONS AND FUTURE WORK

Based upon our review of the literature, we believe that we have identified the core theories required to implement a rigorous, automated, trust framework. Referring to the notional framework diagram from Figure 1 and the various services needed, the following could be implemented:

1. QoI and VoI Services: Use Scruff to reason regarding the hierarchical models for QoI and VoI and provide metadata tagging and annotation services
2. Trust Evaluator: Use the probability of a positive outcome to evaluate trust in sensors and data. Provide mechanisms to discount trust based upon recorded evidence along with memory horizons to "forget" bad experiences
3. Trust Propagator: Synchronize trust among agents by providing recommendations (i.e. gossip) to all agents in the form of opinion tuples on a periodic basis
4. Trust Aggregator: Before and during trust propagation, use trust discounting techniques for combining opinion tuples based upon opinions of others
5. Trust Recorder: Record positive and negative experiences based upon data exchanges (e.g. ICDs) and data processing (e.g. cross-tagging, mis-tagging)
6. Trust Prediction: Use AHP and/or TI-POMPDP to monitor and predict trust based upon the available evidence, current state vector machines, the QoI and VoI metadata, and potential rewards
7. Security Services: Use AHP and/or TI-POMPDP to evaluate decision policies based upon human-on-the-loop context inputs

Once the various microservices within the trust framework have been established, we plan to consider various test cases to illuminate the effect of incorporating trust into traditional SDA processing:

1. Single sensor under our control (no possible disagreement)
2. Single sensor not under our control
3. Multiple sensors under our control (potential disagreement)
4. Multiple sensors out of our control
5. General case

There are open questions that require additional study specific to the SDA problem space. The primary question being how to map complex scenarios like the collision avoidance problem into positive and negative experience tallies. It is conceivable that we could develop a functional relationship between the belief uncertainty and the metric uncertainties along with our knowledge of observation geometries, seeing conditions, sensor reliability, etc.

7.1 Scruff Future Work

Scruff is open source, rapidly evolving, beta research software. Although the software already has a lot of functionality, we intend to expand on this in the future and cannot promise stability of the code or the APIs at the moment. Future work in Scruff will follow five main lines: developing more extensive libraries, including integration of other frameworks; developing a larger suite of algorithms using compositional methods; developing a more flexible framework of networks and recursive models; creating spatial and spatiotemporal models with the same flexibility as current temporal models; and operators for performance characterization and optimization. We welcome contributions from the user community. If any of these items catches your interest, let us know and we will be happy to help with design and development.

7.1.1 Larger libraries and integration of other frameworks

Scruff's current library, particularly of SFuncs, is fairly minimal, and needs to be extended to provide a fully functional probabilistic programming framework. Our intent is not to write sfuncs ourselves, but rather to wrap existing implementations wherever possible. An immediate goal is to wrap Distributions.jl, while will provide a wide range of Dist sfuncs. We also want to integrate with other probabilistic programming frameworks in Julia, such as Gen. In addition, the ability to use data-driven models that don't support sampling but do support inference is central to Scruff. We want to develop a library of such models, again by integrating with existing frameworks and wrapping with appropriate observations. Algorithms also need to be modified to take advantage of such models.

7.1.2 More algorithms

It is important that algorithms in Scruff are well-structured and compositional. The algorithms developed so far are a starter set that have been carefully designed with this philosophy. Noticeable by its absence is MCMC, which is common in many probabilistic programming frameworks. Gibbs sampling can be implemented as a message passing algorithm and fits well with the current framework. Metropolis-Hastings and reversible jump algorithms will take more thought, but experience with other probabilistic programming languages should show how to implement them in a consistent, compositional way.

A very natural next step is to generalize our algorithms to use other semirings besides aum-product. Again, this should happen in a compositional way. It should be possible to say something like with_semiring(semiring, algorithm) and have all computations in operators invoked by the algorithm drawn from the appropriate semiring. If we do this, it will be natural to write learning algorithms like EM and decision-making algorithms using maximum expected utility using our instant algorithms. This will lead to powerful combinations. Would anyone like asynchronous online EM using BP? Similarly, BP is just one example of a variational method. We want to expand BP into a more general compositional variational inference library. Finally, we want to generalize our elimination methods to employ conditioning as well as elimination.

7.1.3 More flexible networks and recursion

The ability for networks to contain other networks is critical to structured, modular, representations as well as efficient inference through encapsulation and conditional compilation. In addition, the ability to generate contained networks stochastically supports open universe modeling. Scruff currently supports these capabilities through Expanders. However, Expanders were an early addition to Scruff and are not integrated all that well in the most recent Scruff development. NetworkSFuncs are better integrated, but do not currently support containment and recursion. We want to align Expanders and NetworkSFuncs to provide more general structured and recursive networks.

7.1.4 Spatially flexible models

Scruff currently has a flexible representation of variables that vary over time, but not of variables that vary over space, or space and time together. We want to provide spatiotemporal networks with the same flexibility as current DynamicNetworks. Moving beyond spatial models, we also want to create a framework for reasoning about variables that vary across graphs, such as social networks.

7.1.5 Performance Characterization and Optimization

Scruff's design is intended to enable reasoning about performance characteristics of operators and to support algorithms making decisions about which operators to use. Multiple operator implementations can exist side by side for given sfuncs and algorithms can use policies to decide which ones to use. This capability is currently only exercised in very rudimentary ways. We want to take advantage of this capability to provide a wide set of performance characteristics and intelligent algorithms that use them.

8. REFERENCES

[1] J. McIntyre, Washington Examiner, 20 January 2022. [Online]. Available: https://www.washingtonexaminer.com/policy/defense-national-security/space-jam-as-earth-orbit-becomes-more-congested-and-contested-critical-satellites-are-at-risk. [Accessed 28 February 2022].

[2] L. Hale, J. C. Last, J. L. De Namur-Paul and R. Barber, Partnering Not Bossing: Better Leveraging International Capabilities for Space Domain Awareness, The Aerospace Corporation, Center for Space Policy and Strategy , August 2021.

[3] J. Stauch and M. Jah, "On the Unscented Schmidt-Kalman Filter Algorithm," *Journal of Guidance, Control, and Dynamics,* vol. 38, no. 1, pp. 117-123, 2014.

[4] U.S. Department of Defense (U.S. DoD), The Dictionary of Military Terms, Joint Pub 1-02, New York: Skyhorse Publishing, 2009, p. 576.

[5] R. Kingsley, Fighting against allies: an examination of "national caveats" within NATO-led International Security Assistnace Force (ISAF) campaign in Afghanistan & their impact on ISAF operational effectiveness, 2002-2012, Massey University Dissertation, 2014.

[6] J. A. Colquitt, B. A. Scott and J. A. LePine, "Trust, trustworthiness, and trust propensity: A meta-analytic test of their unique relationships with risk taking and job performance," *Journal of Applied Psychology,* vol. 92, no. 4, pp. 909-927, 2007.

[7] A. Oltramari and J.-H. Cho, "ComTrustO: Composite Trust-based Ontology Framework for Information and Decision Fusion," in *18th International Conference on Information Fusion*, Washington, D.C., July 6-9, 2015.

[8] B.-J. Chang and S.-L. Kuo, "Markov-Chain Trust Model for Trust-Value Analysis and Key Management in Distributed Multi-cast MANETs," *IEEE Transactions on Vehicular Technology,* vol. 58, no. 4, pp. 1846-1863, May 2009.

[9] K. Govindan and P. Mohapatra, "Trust Computations and Trust Dynamics in Mobile Adhoc Networks: A Survey," *IEEE Communications Surveys & Tutorials,* vol. 14, no. 2, pp. 279-298, 2011.

[10] S. P. Marsh, Formalising Trust as a Computational Concept, Ph.D. Thesis, Dept of Computer Science and Mathematics, Univ. of Stirling,, 1994.

[11] A. Jøsang, Subjective logic, Springer, 2016.

[12] Y. Wang and M. P. Singh, "Formal Trust Model for Multiagent Systems," *IJCAI,* vol. 7, pp. 1551-1556, 2007.

[13] A. Jøsang, "A Logic for Uncertain Probabilities," *Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 9, pp. 279-311, 2001.

[14] A. Jøsang, M. Ivanovska and T. Muller, "Trust Revision for Conflicting Sources," Washington, D.C., 2015.

[15] A. Jøsang, S. Marsh and S. Pope, "Exploring Different Types of Trust Propagation," in *Proceedings of the 4th International Conference on Trust Management (iTrust)*, Pisa, Italy, May 2006.

[16] A. Jøsang and R. Hankin, "Interpretation and Fusion of Hyper Opinions in Subjective Logic," in *Proceedings of the 15th International Conference on Information Fusion*, Singapore, July 2012.

[17] C. Bisdikian, J. Branch, K. Leung and R. Young, "A letter soup for the quality of information in sensor networks," in *Proceedings of the IEEE Information Quality and Quality of Service Workshop (IQ2S'09)*, 2009.

[18] C. Bisdikian, L. Kaplan, M. Srivastava, D. Thornley, D. Verma and Y. R., "Building Principles for a Quality of Information Specification for Sensor Information," in *12th International Conference on Infomation Fustion*, Seattle, WA, July 6-9, 2009.

[19] C. Bisdikian, L. M. Kaplan and M. B. Srivastava, "On the Quality and Value of Information in Sensor Networks," *ACM Transactions on Sensor Networks,* vol. 9, no. 4, p. Article 48, 2013.

[20] K. Chan, K. Marcus, L. Scott and R. Hardy, "Quality of Information Approach to Improving Source Selection in Tactical Networks," in *Proceedings of the 18th International Conference on Information Fustion*, Washington, DC, July 6-9, 2015.

[21] J. Thakkar, Multi-Criteria Decision Making, Springer, 2021.

[22] M. Aruldoss, T. M. Lakshmi and V. P. Venkatesan, "A Survey on Multi-Criteria Decision Making Methods and Its Applications," *American Journal of Information Systems,* vol. 1, no. 1, pp. 31-43, 2013.

[23] T. L. Saaty, The Analytic Hierarchy Process, New York: McGraw-Hill, 1980.

[24] T. J. Saaty, "Decision Making with the Analytic Hierarchy Process," *Int. J. Services Sciences,* vol. 1, no. 1, pp. 83-98, 2008.

[25] L. Vargas, "An Overview of the Analytic Hierachy Process and its Applications," *European Journal of Operaional Research,* vol. 48, pp. 2-8, 1990.

[26] C. A. B. e Costa and J. C. Vansnick, "A Critical Analysis of the Eigenvalue Method Used to Derive Priorities in AHP," *European Journal of Operational Research ,* vol. 187, no. 3, pp. 1422-1428, 2008.

[27] N. Buyurgan and C. Saygin, "Application of the analtyical hierarchy process for real-time scheduling and part routing in advanced manufacturing systems," *Journal of Manufacturing Systems,* vol. 27, pp. 101-110, 2008.

[28] D. S. Bernstein, R. Givan, N. Immerman and S. Zilberstein, "The Complexity of Decentralized Control of Markov Decision Processes," *Mathematics of Operations Research,* vol. 27, pp. 819-840, 2002.

[29] P. Doshi, "A Framework for Optimal Sequential Planning in Multiagent Settings," *Association for the Advancement of Artificial Intelligence,* pp. 985-986, 2004.

[30] R. S. Seymour, The Trust-Based Interactive Partially Observable Markov Decision Process, AFIT. Theses and Dissertations. 2562, 2019.

[31] Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Trust_(social_science). [Accessed 18 June 2024].

[32] L. Kaplan, M. Sensoy and G. de Mel, "Trust Estimation and Fusion of Uncertain Information by Exploiting Consistency," in *17th International Conference on Information Fusion (FUSION)*, 2014.

[33] 18th & 19th Space Defense Squadron, "Spaceflight Safety Handbook for Satellite Operators, Version 1.7," Combined Force Space Component Command, Vandenberg AFB, CA, April 2023.

[34] Office of the Chief Engineer, "NASA Spacecraft Conjunction Assessment and Collision Avoidance Best Practices Handbook," NASA/SP-20230002470 Rev 1, February 2023.

[35] M. P. Wilkins, E. J. Moyer and I. I. Hussein, "Determining Type I and Type II Errors when Applying Information Theoretic Change Detection Metrics for Data Association and Space Situational Awareness," in *Advanced Maui Optical and Space Surveillance Technologies Conference*, Wailea, HI, 2017.

[36] G. L. Rogova and E. Bosse, "Information Quality in Information Fusion," in *13th International Conference on Information Fusion*, 2010.

[37] B. E. Ruttenberg, M. P. Wilkins and A. Pfeffer, "Reasoning on Resident Space Object Hierarchies Using Probabilistic Programming," in *18th International Conference on Information Fusion*, Washington, D.C., July 6-9, 2015.

[38] Charles River Analytics, "Scruff," [Online]. Available: https://github.com/charles-river-analytics/Scruff.jl. [Accessed August 2024].