Utility and cost evaluation for shared awareness

van Foeken, E.

Utility and Cost evaluation for Shared Awareness

E. van Foeken

Utility and Cost evaluation for Shared Awareness
E. van Foeken

Utility and Cost evaluation for Shared Awareness

E. van Foeken

Universiteit van Amsterdam (UvA)

Utility and Cost evaluation for Shared Awareness

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan
de Universiteit van Amsterdam,
op gezag van de Rector Magnificus
prof.dr. ir. K.I.J. Maex
in het openbaar te verdedigen
ten overstaan van een door het College voor Promoties ingestelde
commissie, in het openbaar te verdedigen in de Agnietenkapel
op woensdag 25 oktober 2017, te 10 uur
door **Eelke van Foeken**
geboren te Groningen

Promotiecommissie

Promotor:	Prof. dr. ir. F.C.A. Groen	Universiteit van Amsterdam
Copromotor:	Dr. ir. L.J.H.M. Kester	TNO

Overige Leden

prof. dr. ir. C.T.A.M. de Laat	Universiteit van Amsterdam
prof. dr. P.W. Adriaans	Universiteit van Amsterdam
prof. dr. J.L. van den Berg	Universiteit Twente
prof. dr. ir. E.R. Fledderus	Technische Universiteit Eindhoven
dr. A. Visser	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

This research has been performed at the Distributed Sensor Systems group of TNO and the Intelligent Autonomous Systems group of the University of Amsterdam.

Contents

Chapter 1

Introduction

The human brain collects information comparable to billions of bits of raw data from the eye, skin, ear, nose, and mouth every second, but its' consciousness can process a mere 40 bits per second (Pradeep [2010]). Besides information abstraction, humans concentrate on the relevant parts of the environment by using a cognitive process named *attention*. Humans allocate attention resources to those stimuli that are most relevant for achieving their current goals. A striking and famous example is the *Cocktail Party Effect* where a person can focus on a certain voice in a loud room.

One part of *visual* attention can be modeled through the zoom-lens model (Eriksen and James [1986]). This model describes that, besides the position of the focus of attention, the size of the focus area can change as well. Interestingly, this model describes a trade-off between the size of the focus area and the efficiency of processing (Castiello and Umilta [1990]). Regardless of the focus area size, a fixed amount of attention resources is available to process the visual information. Hence, the larger the area the slower the processing of that area. This trade-off may be compared with a trade-off between the *value* of the data within the focus area and the *cost* of processing that data.

The real-time process of filtering data and balancing the *value* and the *cost* of data is exactly what we want in an artificial counterpart that is a *Distributed Sensor System*, which is a system consisting of multiple spatially separated *entities*. An entity in this thesis has at least one sensor, where a sensor has an ability to observe the environment. Spatial distribution of sensors is enforced through two *constraints* situated in the system and its environment. Firstly, the limited range and resolution of sensors necessitates physically distributed sensors, especially in large-scale environments. Secondly, sensors are in close interaction with the environment and thereby more vulnerable to failure or breakdown. A redundancy of sensors will therefore result in higher *robustness*.

A *Distributed Sensor System* can be applied in many domains, such as the civil and the military domain. Within the security and safety domain in Hofmann and Gavrila [2002] a system is presented that can estimate the 3D upper body movement from multiple camera's. Another example is from Scerri et al. [2007], were

Unmanned Air Vehicles (a.k.a. UAVs) collectively use sensors that can measure the Received Signal Strength to find the location of objects emitting Radio Frequency signals, with applications ranging from military to civilian, like finding lost hikers. In Kasteren et al. [2010] multiple different sensors are used to monitor the activity of seniors in their homes. Activities, like opening and closing of doors, sitting or laying in bed, moving of objects like drawers and toilet behavior are measured with uninstrusive sensors (*i.e.* not camera's). The leading example of this thesis is part of the military domain. It concerns a *Distributed Sensor System* positioned on a group of ships which use their radars to surveil the objects in their combined visible environment.

As with humans, the *Distributed Sensor System* aims to construct and maintain awareness of the situation in the environment—*situation awareness*, and plan and execute actions based on the *situation awareness*. Analogously to humans, the amount of information collected though the sensors of a *Distributed Sensor System* (like camera's, radars, infrared sensor and microphones) is significantly larger than the amount of information in the eventual constructed *situation awareness*. Next to the resources for processing and storing that occur in humans as well, resources for communicating data are needed in *Distributed Sensor Systems* to share data between entities. These resources are constrained when timely action towards perceived objects is required. This thesis focusses its attention on dealing with limited communication resources because they are most severely constrained in the application domain under discussion.

First of all, communication constraints are most severe because complex sensors produce vast quantities of complex data that are bound to overload the communication channel. Moreover, in the future it is to be expected that *communication performance* in wireless and ether-networks will increasingly fall behind the *processing performance* of single entities. In some applications, the amount of data to be communicated will increase even more due to the fact that sharing data will be done on a lower level of *information abstraction*.

Higher abstraction levels of information occur in human information processing of sensor data, where raw data is incrementally processed into progressively higher information abstractions. In humans, the cognitive model by Fuster [2004] describes a hierarchy of information abstraction levels for perception in the posterior cortex. These levels operate at increasingly higher abstraction levels of information. Higher level information is constructed by fusing lower level information and discarding certain details.

In a *Distributed Sensor System* a hierarchy of information abstraction levels can be modeled as well, and one such hierarchy is suggested by
(of the Joint Directors of Laboratories [1991]):

3 Impact Assessment

2 Situation Assessment

1 Object Assessment

0 Signal Assessment

Sensors collect raw data that can be progressively processed into signal, object, situation and impact information. As details are discarded with each abstraction step, the amount of information is decreased. Because the entities harboring sensors are distributed in space, communication is needed when they need *shared awareness* of the environment. Shared awareness can be required on every level. However, on a lower abstraction level there is more information that can be shared.

The maritime operations that are exemplified in this thesis can benefit from sharing awareness on a lower abstraction level than is used presently. If we look at Fig 1.1 three ships are shown that have a shared awareness. Each ship has the same information abstraction hierarchy. Nowadays the construction of *shared awareness* is entity centric, *i.e.* each entity constructs *situation awareness* on its own and only when it is more or less complete will it be exchanged through classical data links in the form of tracks. In the figure the datalinks would be between the situation assessment level components on each ship.

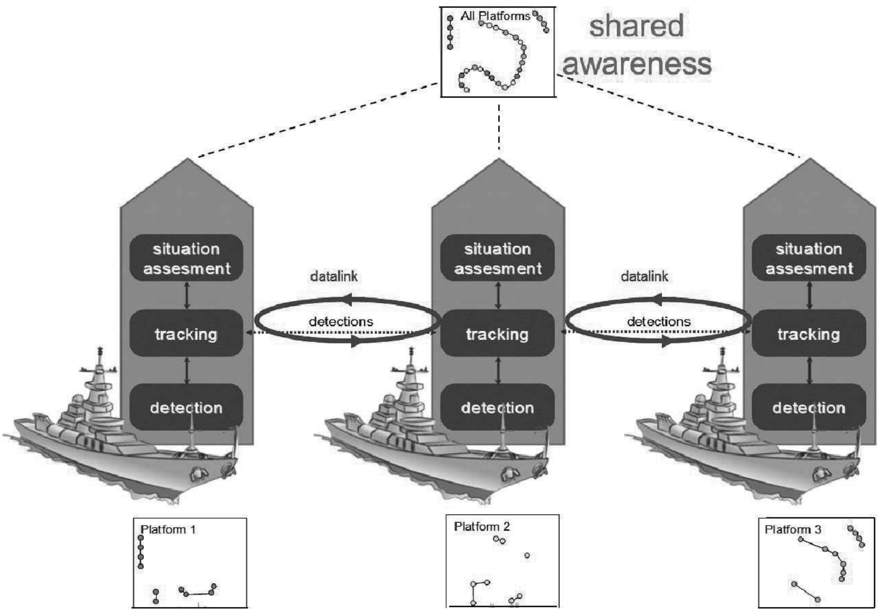

Figure 1.1: Each ship has information abstraction levels and the abstraction levels are spatially distributed. Detections are shared on the object assessment level and an *Identical Shared Awareness* is created from them.

The top half of Fig. 1.2 demonstrates what the tracks will look like in this case. Each ship uses its local plots—plot is the name for a detection or sensor reading in the maritime domain—to create a local track of the object it is observing. After a fixed time-interval the ships exchange their local tracks—and note that these are different from eachother—so that the ships have several local tracks of the object, see the *integrated picture of tracks* on the far top right of the figure.

This methodology works for missions on the high seas but coastal operations are proven to be more complex. Tracking of objects can be much harder when ships are operating near the seashore because of possible things blocking the line-of-sight, like buildings or mountains. When there is no line-of-sight for a while, gaps in the local track can occur, or worse, the track cannot be continued anymore once the ship has line-of-sight again. Moreover, coastal areas have a higher chance of busier air and sea traffic, making it harder to track the relevant objects. We argue that by fusing information on an abstraction level lower than track information, the chances of detecting objects can be higher and the accuracy of the kinematic estimation and track continuity be improved.

Therefore, considerable improvement of the quality of the *shared awareness* can be achieved by exchanging detections instead of tracks, followed by fusing the detections of the different ships into a composite track. The lower part of Fig. 1.2 reflects this process; detections from different ships are fused into one track. When multiple ships have an object in view the chance of detection is higher, because the frequency of detections is higher. Increased frequency also brings a higher accuracy of a track. Chances on track continuity are also higher because the different points of view of multiple ships are used. One ship may not see the object but receives detections from two other ships that do see the object. This can keep the track continue without gaps.

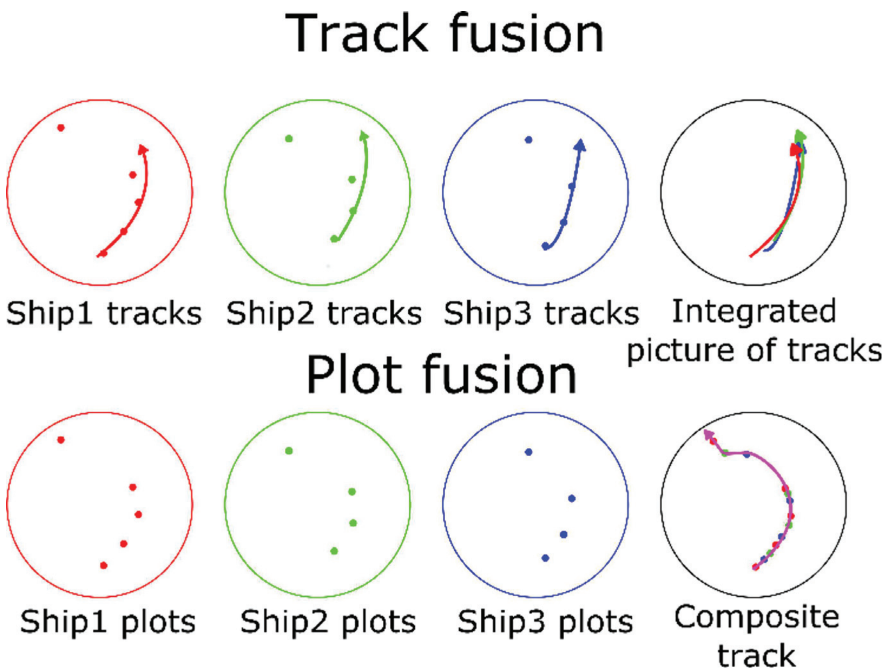

Figure 1.2: Top: three ships using classical data-links to exchange complete tracks. Bottom: same three ships sharing detections.

Despite the many advantages, there is more information to be shared on lower abstraction levels—*e.g.* a single track is smaller in size than the 10 detections comprising it. This requires more bandwidth and can cause significant delays in communication. In case timeliness of *shared awareness* is required, flexibility in communication and run-time selection of relevant information may increase the adaptivity to communication limitations and enhance the quality of the *shared awareness*.

In this thesis such flexibility is developed for a *Distributed Sensor System* that constructs and maintains an identical and shared awareness—*identical shared awareness*—for every object in the visible environment, see Fig. 1.1. *Identical shared awareness* is defined as an estimated state representing the relevant features of a currently relevant object in their combined visible environment that must meet the following demands:

identical the state must be identical on all entities. Therefore, information that is intended to be used must be shared and must be processed by the same methods.

synchronized the state must be time-synchronized on all entities, hence must be equal at all times. This means all entities must first have confirmation that the relevant information has arrived at the designated entities, to be followed by a simultaneous update of the *identical shared awareness* with that information.

For *identical shared awareness* to be useful some additional assumptions have to be made:

- The entities know from each other that they have *identical shared awareness*
- The requirement to coordinate action plans on the *identical shared awareness*

The advantages are therefore that the entities can coordinate their behavior towards objects without any additional communication which makes the reaction time shorter. Moreover the combined effect of multiple entities is higher on an object then when these entities act individually.

Let's address a purely hypothetical example: there are three ships observing their combined visual field. There is a single hostile object entering the visual field. The entities observe that it is hostile and know this from each other. At a certain point in time they have high enough accuracy of the location of the object to engage their predefined simultaneous reaction to eliminate the object with their weapon systems. Chances of elimination increase due to the higher number of entities in action.

The object assessment level has to share data between entities, but as indicated, sharing this data is hampered by the communication network that has insufficient resources to timely communicate all data. On the one hand, sharing information brings a certain *cost* of communication; there is both a delay and a required amount of resources. The delay is caused not only by sending the message but also by the time it takes for all the entities to have confirmation of the reception of the message by all entities. On the other hand, information brings a certain *value*.

We relate the value of information for the *identical shared awareness* indirectly to the goals the *Distributed Sensor System* has towards the objects in the environment, just like humans often focus their attention on something that is valuable for their decision-making. In this case we assume the higher information abstraction level constructs *information-requests* that indicate what are the currently important features of information.

The methods presented in this thesis are aimed to balance the *value* and the *cost* of information in order to improve the quality of the *identical shared awareness* by satisfying the goal-directed *information-requests* and, at the same time, not overly use the available communication resources.

Multiple aspects are involved in improving the *identical shared awareness*:

1. the communication technique,
2. the run-time evaluation of communication capabilities,
3. the evaluation of the run-time contribution of information to the *identical shared awareness* given the *information-requests* and
4. the combination of evaluating data and communication.

The evaluation methods, of which their mechanisms are the main contributions of this thesis, improve with better communication techniques and estimation of the current communication performance.

1.1 Communication Model

Reliable and controlled wireless information sharing in a network centric type approach is needed for achieving timely *identical shared awareness*. These techniques can especially help when complex sensors are involved that generate large amounts of data in complex and dynamic environments, where communication constraints—for example bandwidth and latency—pose significant limitations.

To enable reliable experimentation with such advanced techniques in real-world applications or in simulation we have developed a communication model that is realistic to the extent that it describes the most important performance indicators of wireless communication systems, such as *link stability*, *throughput* and *latency*, but does not use complex channel models, to be generic for varying communication techniques in a wide variety of scenarios.

Furthermore, the communication capabilities influence how well information is shared. Especially delay is of huge influence on how timely and accurate the *identical shared awareness* will be. Also the amount of communication resources used is of importance. The costs of communication increase when more resources are used to transmit information across the network. The idea of the evaluation methods is that they adapt, at run-time, the communication to the varying *cost* due to changing communication capabilities, and the varying *value* of information, which is influenced by the delay.

Therefore, the communication model needs to be able to evaluate the *expected delay distribution* and the *expected cost of communication* at every instance. In the literature it is hard to find suitable generic, low-complex models and therefore one of the contributions of this thesis is a model that can evaluate the *expected delay distribution* and the *expected cost of communication*.

The low-complexity model derives the *expected delay distribution* and *expected cost of communication* from a certain setting with parameters like power, frame-time and bandwidth, but also environmental parameters such as distance between entities and roughness of the ocean. The *Communication Service* provides updates of changes in the real-time communication capabilities, so that the evaluation methods can adapt to these changes. In addition, within the parameter boundaries of the used communication system, the evaluation methods can also use resource management to allocate resources to improve the *expected delay distribution* and *expected cost of communication*. For example, resource management can re-allocate resources, like adding more power for transmitting a certain message, to change from an unacceptable to an acceptable expected delay. This model determines the probability of latency in terms of the *expected delay distribution* for multicast transmissions. It also determines the probability of the required resources in terms of the *expected cost of communication*.

1.2 Utility and Value of Information

The third aspect is to improve the evaluation of the run-time contribution of information to the *identical shared awareness* given the *information-requests*. We use *information-request*s to steer the construction of *identical shared awareness* in the desired direction. The nature of a certain *information-request* depends on the *domain*, the *information abstraction level*, the *goals* and the current *situation*. The domain here is *Distributed Sensor System*s that perform evaluation on the 'object' level of information abstraction. Therefore, the domain and level are fixed, but the goals and the situation may change during a mission. The goals of the system are known to request a certain effect on the perceived objects. Depending on the current requested effect, certain features and certain objects will be more important to be observed than others, and this will be reflected in the current *information-request*.

It is important what characteristics an *information-request* should minimally have. First of all, an *information-request* posed by a higher information abstraction level needs to have a clear relation with the type of information that the lower information abstraction level produces. The request should quantitatively indicate what and/or when information gathered by the lower abstraction level is relevant. We formulate the *information-request* in the form of a *utility function*:

> The utility function describes the quantitative utility of the important features of information.

The purpose of a *utility function* is to reflect the current *information-requests* for certain features of the *identical shared awareness*, such as accuracy and timeliness. As mentioned before, but important to emphasize, these *information-requests* are driven by the aimed effect the *Distributed Sensor System* wants to have on objects in the environment. For example, when the weapon system of a ship needs at least a location accuracy of $5\ m$ of hostile objects the *utility function* will be a step function where the step is at $5\ m$—Fig. 1.3.

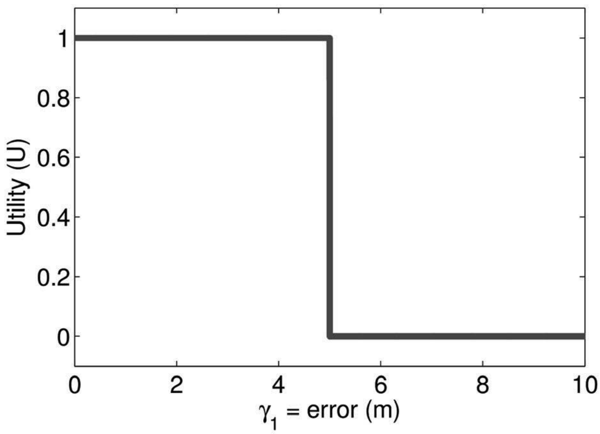

Figure 1.3: An example of a *utility function*.
It represents a step function, where an error of tracks of less than 5 meters is useful, and an error of more than 5 meters useless.

Given the utility function, the entities on the lower information abstraction level can calculate the *value* of newly gathered data. *Value* is defined as the *utility* that is gained from using the data for the *identical shared awareness*. The gain is defined as the difference between the *utility* of the *identical shared awareness* before the data is used and the *utility* after the data is used. This thesis offers a novel function that calculates the *utility* of information. We stress that the evaluation methods presented in this thesis can also use other definitions of the *utility* of information. However, when the quantitative *utility* of features derived from *probability density functions* (signifying the uncertainty of the estimation) are required, such as the estimate stated of the tracks of objects in this thesis, the *utility* can be *more directly* measured in relation to *information-requests* in comparison with methods that use information-theoretic measures. The improved *integrated utility function* overcomes implicit features attributed to information-theoretic measures such as information divergence.

Ultimately, in addition to evaluating the contribution of information to the *information requests*, evaluating the *expected cost of communication* simultaneously best improves the *identical shared awareness* by a communication constrained *Distributed Sensor System*.

Therefore, in this thesis evaluation methods are presented that optimize the *identical shared awareness*, by taking into account both the *expected value* as well as the *expected cost of communication*. They adapt the exchange of local information to the possibly changing *information requests* and possibly changing communication capabilities.

The first evaluation algorithm presented in this thesis is *Request and Constraint Based Evaluation*. *Request and Constraint Based Evaluation* decides whether information has enough *expected reward* to communicate to the other entities for improving the *identical shared awareness*. The *expected reward* is the difference between the *expected value* and the *expected cost of communication*. If it is positive the method considers the information relevant and decides to communicate it. This *expected reward* is based on the current *information-requests* and communication capabilities. Information can be rewarding based on one *information-request* and unrewarding on another, or rewarding based on good communication capabilities and unrewarding on bad ones.

The current communication capabilities are estimated by the *Communication Service* in the form of an *expected delay distribution* and an *expected cost of communication*. The *expected delay distribution* is part of both the *expected value* and *expected cost of communication*. *Request and Constraint Based Evaluation* enhances the adaptivity to *information requests* and constraints. The novelty of the method lies in the *combination* of both *information-request* evaluation at run-time and constraint evaluation of information at run-time.

The second evaluation algorithm is an extension on *Request and Constraint Based Evaluation* and is called *Adaptive Team Formation*. Where *Request and Constraint Based Evaluation* run-time determines whether each new collected information is valuable enough to share with the other entities in a team, *Adaptive Team Formation* extends *Request and Constraint Based Evaluation* by dynamically determining which entities should be part of the team. *Adaptive Team Formation* calculates the contribution of an entity based on the quality of features such as the sensors, the communication capabilities and the utility of its data for the *identical shared awareness*. Shortly, *Request and Constraint Based Evaluation* determines whether to share and *Adaptive Team Formation* determines whom to share with.

1.3 Objectives and organization of the thesis

The general objective of this thesis is:

> To provide innovative methods for creating the *best shared awareness* of objects in a dynamic environment within a communication constrained distributed sensor system with varying goals.

This results in the following objectives:

- Determine how to best relate the information-requests/utility functions of the higher information abstraction level to the information that the lower abstraction level produces,

- Simultaneously deal with varying information-requests and limited communication capabilities,

- Estimate the run-time communication capabilities,
- Determine the *utility* and *value* of information,
- Determine the contribution of entities to the *identical shared awareness*.

These objectives we try to attain through this thesis. The thesis consists of 5 essential chapters.

Chapter 2 describes the state-of-art and motivates the choices made. The choices that are made regarding the architecture for designing *Distributed Sensor Systems*, regarding the how and why of maintaining an *identical shared awareness* and regarding the communication model are motivated. Also, argumentation is given for the use of the *integrated utility function* in relation to existing literature. In addition, the evaluation methods are compared with earlier work and their contribution is presented. Lastly, we argue that performance can better be measured by using measures that relate more directly to the *information requests*.

Chapter 3 gives a detailed account on the communication model that can estimate the run-time communication capabilities in the form of an *expected delay distribution* and an *expected cost of communication*. It also includes examples to show the advantage of the model for evaluation methods.

Chapter 4 discusses and motivates the novel calculation of the utility of information and is named the *integrated utility function*. Its advantage over other utility-calculation approaches is shown. The *integrated utility function* and the communication model are essential parts of the evaluation methods.

Chapter 5 presents the *Request and Constraint Based Evaluation* method. Through simulation experiments in two maritime scenarios we show the possibilities of this method in general, of the communication model and of *integrated utility function*.

Chapter 6 comprises the *Adaptive Team Formation* method. Through simulation experiments its benefits over *Request and Constraint Based Evaluation* is shown as well. It can result in dynamic team compositions under dynamic circumstances in *information-requests*, communication capabilities or environment. The final chapter is about the conclusions and future work.

Chapter 2

State–of–the–Art

In this chapter we review the state-of-the-art on the main ideas presented in this thesis.

First of all, we have to design an architecture for a *Distributed Sensor System* that can construct an *identical shared awareness*—ISA that is adaptive to the *information-requests* and *communication capabilities* of the entities. Many features for such an architecture are already presented by Steinberg and Bowman [2004] and Kester [2010], such as *information abstraction* and *provider-consumer* interactions, and we take these features along in our architecture. But, features such as the explicit distinction of design-time from run-time modeling or detailed interface and interaction definitions need to be added.

With regard to ISA there are some definitions that relate to our concept of ISA, but a novel definition that combines these definitions need to be made. In Dorion and Boury-Brisset [1998], several possible levels of inter-operability between entities are described. Which level do the entities in our DSS need to construct and maintain ISA? And how does the comprehensive shared awareness in Kingston and Martell [2004] compare with the concept of ISA here?

Another important aspect is the modeling of communication. The state-of-the-art in this subject offers a rich variety of such models. However, the goal of our model is to be able to evaluate the delay and cost of communication given the used communication technique and the current scenario that the DSS is placed in. The state-of-the-art either provides too detailed models or too generic models for this purpose. Therefore, we needed to develop a model that fits our needs and is not too generic but also not too detailed.

There are many different uses of the terms utility and value of information in literature. Therefore it is essential to investigate these and introduce our own definitions. In Eswaran et al. [2011] and Velagapudi et al. [2007] we found useful theoretical features of *utility* and *value*. The first more from a philosophical point of view, since it answers the question of what utility should be defined like in networked and goal-driven systems. The second more from a mathematical point of view in calculating the *value* of information. Our own definitions of *utility* and *value* are mainly inspired from these two references. An important issue that we need to address was the *indirectness* or *implicitness* of the information theoretical measures, such as Kullback Leibler divergence, often used in multi-agent research such as Velagapudi et al. [2007] but also in many sensor management techniques, Grocholsky [2002].

Ultimately it is our goal to be both adaptive to communication costs and information utility or value. Many solutions have been offered regarding this issue. However, without looking at the exact details of how these solutions value information or deal with communication issues, these solutions always miss a certain feature. The one does perform information evaluation but is not adaptive to communication constraints Velagapudi et al. [2007], where another does evaluate the current communication capabilities but is not adaptive to information-requests, for example the resource management technique. These methods form the starting point to our first evaluation method, RCBE, of this thesis.

The second evaluation method goes one step further and is inspired on methods that adaptively reconfigure the entities that cooperate in teams to achieve goals (Bolderheij et al. [2005], Spaan and Lima [2009], Howard and D. Payton [2002]). *Adaptive Team Formation*—ATF—requires evaluation of entities based on their contribution of information to the ISA and their communication capabilities. Especially the last two offer interesting solutions to dynamic team formation.

The last part of the chapter discusses the performance measures that we believe should be used in systems with dynamic mission-goals, the demonstration environment of the experiments and the generic formulation of the DSS used in the experiments.

2.1 Architecture

To address the problem of creating the *best shared awareness* of objects in a dynamic environment within a communication constrained DSS with varying goals, we require an appropriate architecture for designing such a DSS. Such an architecture provides guidelines in designing such a DSS.

The main characteristics that our architecture should accommodate are:

design-time vs run-time modeling our system has parts that are determined at design-time but the parts that are performed at run-time is the focus on in this thesis.

physical decomposition of the system our system consists of multiple sensors deployed in space.

information abstraction decomposition the system fuses low level detection information into higher level track information.

interface definitions between components between components of the system interfaces need to be defined.

interaction definitions between components what are the the actual interactions that will take place between components. By providing these features a system can be designed that is distributed, has several information abstraction levels, has interfaces between entities as well as between components on single entities and that accommodates the installment of evaluation methods on components that act at run-time.

We review the characteristics in three sections:

(a) design-time versus run-time modeling,

(b) decomposition: physical and information abstraction

(c) interfacing: interface configuration and interaction refinement

Based on these five characteristics similar architectures are compared. As a starting point we take two architectures described in Steinberg and Bowman [2004] and Kester [2010], augmenting them with the specific features needed for our DSS.

2.1.1 (a) Design-time vs. Run-time

To begin with, in designing systems we stress the importance to distinguish the properties or behaviors that are determined at *design-time*—prior to the launching of the system—from those properties and behaviors that are determined at *run-time*—during execution of the system. On the one hand designers can totally predefine or hard-wire a system by determining all properties and behaviors at design-time. It follows that the entities and processes in the system know *a-priori* what to do. On the other hand the system can define all properties and behaviors at run-time: the system has to learn how to configure itself without any prior knowledge and without supervision. These two extremes demonstrate the least adaptive and the most adaptive system respectively.

This is important because it can be fruitful to determine beforehand what is done at design-time and what is done at run-time. Firstly, it can give the designer a clear overview of the run-time and design-time parts. Our proposed RCBE system adapts in run-time the sharing of information with a design-time determined team of entities, ATF determines the teams at run-time. If another evaluation method is considered, the designer can beforehand have a quick comparison with other evaluation methods which parts are done at design-time and which in run-time. Secondly, important foreseen and unforeseen issues arise for run-time determination that do not occur in design-time determination. The simpler issue is to determine the run-time reaction to foreseen situations, like the turning on reaction of a thermostat when its colder than the wished temperature. However, it is harder to react to unforseen circumstances. The first issue can be solved by pre-programming the adaptivity of the system to the foreseen circumstances. And our system falls in this category. The second case were unforseen situations arise are out of the scope of this thesis (topic of human agent systems, learning systems).

In Steinberg and Bowman [2004] and Kester [2010] architectures are presented for designing systems that have partly design-time and partly run-time determined properties and behaviors. However, there is no explicit observation that it can be fruitful to determine beforehand what is done at design-time and what is done at run-time. Compare for example the design of a thermostat, where everything is done at design-time and a cloud computing application Armbrust et al. [2009], where the amount of memory, processing and communication resources is determined at run-time based on the users request. The difference in how much is configured at run-time is significant.

As well as a thermostat or cloud computing application one DSS can differ from another in which properties and behaviors are determined at *design-time* and at *run-time*. For example, in Gulrez and Kavakli [2007] a DSS consisting of hundreds of video cameras, infrared and laser sensors is physically allocated at *design-time*. Two categories are mentioned: one where it is at *design-time* determined that the sensors *continuously* relay their information to a base station, and the other where the system *run-time* determines which sensors communicate information based on the current user query.

The trend is that systems perform more and more processes at *run-time*—look at cloud computing for example. This trend is depicted by the NAIHS evolutionary interaction model in Kester [2010] that shows a road-map for the capabilities of networked systems.

de-conflict entities operate independently but communicate as not to conflict each other.

coordinate coordination of tasks between entities, but still with very limited interaction.

integrate capabilities adjust their services to changing situations.

coherent (also known as Agile) the whole system re-organizes itself to changing situations.

The two evaluation methods presented in this thesis fit nicely in this list; RCBE adjusts the sharing of information with a team of entities to the changing *information-requests* and *communication capabilities* corresponding to the *integrate* phase; ATF re-organizes the team of entities tracking an object to the changing *information-requests* and *communication capabilities* corresponding to the *coherent* phase.

2.1.2 (b) Decomposition

We require an architecture that enables two decomposition principles: *information abstraction* and *multiplicity or parallelism due to—physical—constraints*. Information abstraction enables different levels of information and the possibility to install interfaces between levels. These interfaces enable a higher abstraction level to pose *information-requests* to the lower abstraction level. *Multiplicity or parallelism due to—physical—constraints* is essential for distributed sensor systems, with the additional advantages that a spread of sensors can improve the ISA in, for example, robustness, detection range, accuracy and track continuity.

Regarding the *information abstraction* principle it is important to find good abstraction levels. One set of progressively higher abstraction levels is suggested by Steinberg and Bowman [2004]:

3 Impact Assessment: Assessing the impact of the situation on the goal of the system

2 Situation Assessment: Assessing the relations between objects seen in the environment

1 Object Assessment: Assessing the features of an object, such as trajectory, identity, intention

0 Signal Assessment: Assessing the signals observed by a radar, camera or other sensors.

Information abstraction can occur in a DSS if low-level data is iteratively processed into progressively abstract information. For example, sensor systems that process raw data into signals—level 0, signals into object detections—level 1, detections into object tracks—level 2, and finally tracks into a picture of the situation with relations between objects—level 3.

This hierarchy is a decomposition of the *create situation awareness* part of a system. The *cognitive* Networked Adaptive Interactive Hybrid Systems—NAIHS—model by Kester [2010] and the Real-time Control System by Albus and Barbera [2005] use similar decompositions, with differences in the number of levels and the exact nature of the levels.

In this research, the goals that a DSS has towards the objects in the environment are related to the value of information for the ISA. In other words, the actions that a DSS wishes to take are dependent of the ISA at hand: observation is coupled to action. To enable the design of features that involve the actions of a DSS we also require a similar abstraction hierarchy on the action side.

In Steinberg and Bowman [2004] and Kester [2010] a *decide on action* part is coupled to the *create situation awareness* part to constitute the 'brain' of the DSS. Like the *create situation awareness* part the *decide on action* part can further be decomposed in a hierarchy of functionalities. The chain from collectors to effectors is pictured by the well-known Observe Orient Decide Act (OODA) loop Boyd [1987]; Observe and Orient on the collector side; Decide and Act on the effector side.

To each assessment level, as suggested by Steinberg and Bowman [2004], a *decide on action* or *management* level can be coupled, which generates intentions based on the generated situation awareness. In short, information runs, like in the brain, through a chain of connected hierarchical levels of perception and action that constitutes the perception-action cycle. Although the action part of a sensor-actuator system is essential, the focus in this thesis lies primarily on the architecture to *create situation awarenesst*.

The second principle for decomposing the system is *multiplicity or parallelism due to—physical—constraints*. Due to physical limitations of the sensors and redundancy of observations, it may be advantageous to spatially decompose the system into multiple entities. Each entity can then have their own information abstraction hierarchy and abstraction levels can be split over entities.

Besides *information abstraction* and *parallelism* as principles one other principle is present as well: *time abstraction*. We disregard this abstraction since we assume information abstraction and time abstraction coincide and we do not use time abstractions explicitly in this thesis. We assume that the higher the abstraction level is the higher the time abstraction will be. One of the novelties in Kester [2010] compared to Steinberg and Bowman [2004], Albus and Barbera [2005] is that it is stressed that these three principles can be loosely applied, meaning that a system can be decomposed along any subset of these principles. This loose application of principles makes the architecture more generic than the other architectures just mentioned.

The Real-time Control System by Albus and Barbera [2005] is an example of an architecture that makes similar decompositions. Here and in Steinberg and Bowman [2004] as well, instead of a loose application of the principles, a strict coupling of information abstraction and space-time abstraction is suggested; if the area of observation increases the information becomes more complex and the planning is done for increasing intervals of time. For example, in Albus and Barbera [2005] there is a level where 500 m maps are used explicitly for vehicles that plan actions within a 50 second time-frame and on immediate objects of attention. Another level corresponds with a platoon of vehicles acting in a 50 km range and make plans in a time-frame for the next 2 hours and relative to distant objects. Although this might be a good mapping of information abstraction and time abstraction in one application, one could imagine other mappings for other applications. For instance, one could require a 50 second time-frame for the 50 km range as well. The architecture in Kester [2010] grants the flexibility to do so.

For example, a system could be working on information about single objects—a single information abstraction level— but integrate object information on different time-scales, like every millisecond, but also every minute and even every hour. Most swarming applications are dealing with multiple entities with identical behavior or a common interaction mechanism, like in Gaudiano et al. [2003] where multiple Unmanned Air Vehicles (UAVs) with equal capabilities are dispersed to surveil an environment. They mostly act on the same information and time abstraction level and follow simple rules. In the AI domain Brooks [1991], who discards the decomposition in information abstraction, proposes a hierarchical composition of process cycles based on reaction or cycle time. In conclusion, any system can be defined along these three principles; from a simple to a highly complex system.

A minimal view of our system is depicted in Fig 2.1. For now it is only important to recognize the distribution of entities where each entity has its own levels of information abstraction. The four information abstraction levels from Steinberg and Bowman [2004] are represented in horizontal order from left to right. Red and green indicate the parts that are determined at *design-time* and *run-time*, respectively. This is just an example of our system, and which parts are determined at design-time and run-time is application dependent.

This two-dimensional decomposition leads to 'functional' *agents* acting on a certain information abstraction level and on a certain entity. Note that 'functional' *agents* need to be distinguished from 'physical' agents; a 'physical' agent is an entity that has a physical individuality which perceives with physical sensors and acts through actuators and a 'functional' agent can have any form or shape as long as it has a certain functionality. Since there are multiple agents on a single physical

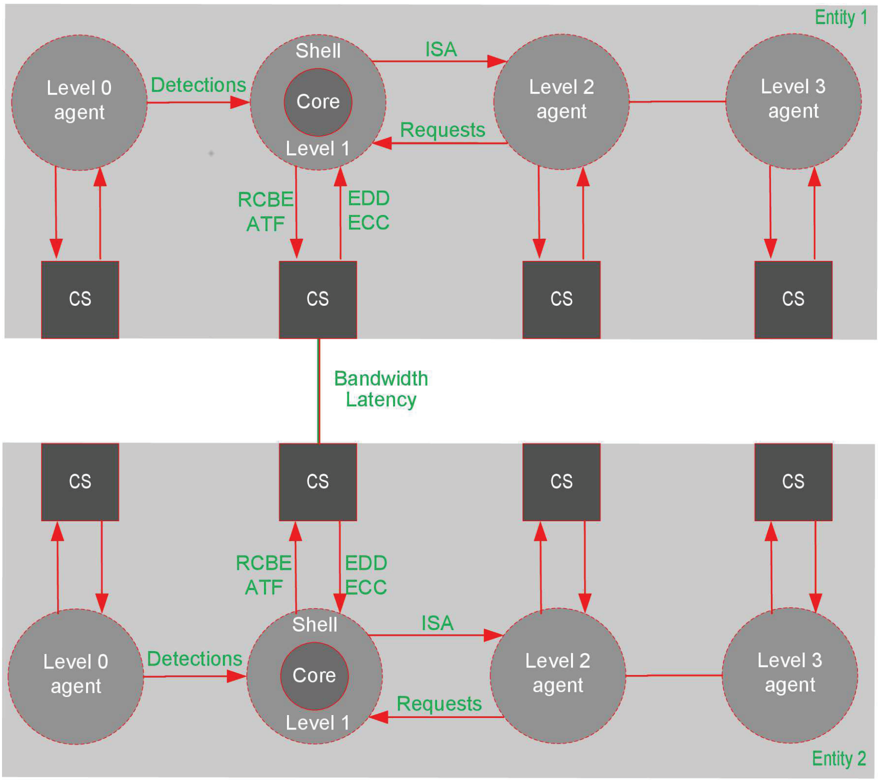

Figure 2.1: A generic layout of the important features and modeling steps of an adaptive networked system. Red indicates the properties that are determined at design-time and green those that are determined at run-time. At design-time the system is decomposed into successive information abstraction levels and decomposed into physically distributed entities. In addition, the interfaces are set between successive levels and between the *Communication Service* (CS) and the agents. These agents are beforehand equipped with a *Core* and a *Shell*. In our DSS, level 2 agents determine at run-time their information-requests and the level 1 agents deliver information accordingly.

entity we are dealing with 'functional' *agents*.

The distributed entities can act separately, but as the road-map of Kester [2010] suggests, systems are and will become more and more connected and coordinated. The architecture therefore needs to account for communication functionalities. Each entity is therefore equipped with services that function as a Communication Service—CS. Each *agent* can transmit information through its CS.

In retrospect we can define the following generic functionalities of a DSS in Fig. 2.2. A DSS has *Collectors*/sensors that collect information, *Effectors* that affect the environment, a *Create Situation Awareness* and a *Decide On Action* part, where the spatially decomposed system is connected through a *Network*. *Communication* is possible between entities over the *Network* and *Computation* is needed to process information into higher information abstraction levels and to transmit information.

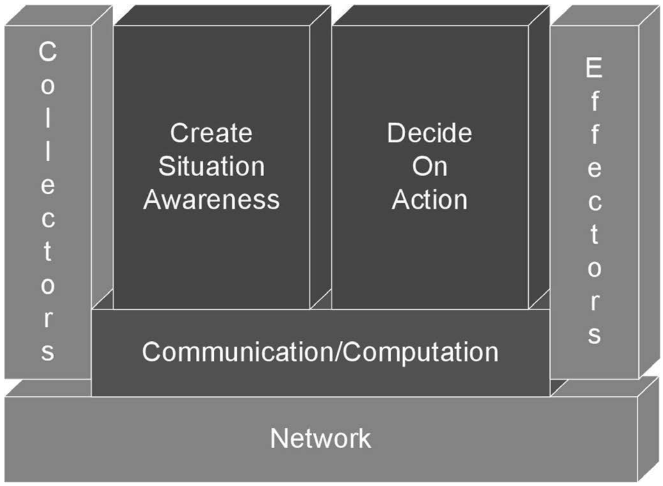

Figure 2.2: The first functional partition of a system according to Kester [2010] into a hybrid mind (create situation awareness and decide on action), collectors (collecting data from the environment) and effectors (acting on the environment) and a computation/communication functionality.

When the separate components have been determined, it may be valuable to determine which components are interfaced and how the interactions between components should be. We distinguish therefore two phases: the *Interaction Configuration Phase* determines which components are interfaced and what type of interaction they perform and the *Interaction Refinement Phase* determines how the interactions run in detail. This partition makes the design process more modular and clear compared to combining the phases into a single phase. The modularity also makes it easier to distinguish design-time from run-time determined parts of the DSS.

2.1.3 (c) Interfacing

Interface Configuration Phase (ICP)

Our architecture needs to account for interfaces between components, since entities need to be able to share information and a higher abstraction level agent and a lower abstraction level agent need to be able to communicate with each other bidirectionally. In Albus and Barbera [2005], Boyd [1987] interfaces are implemented between successive levels in the information abstraction hierarchy. These are, however, unidirectional. Steinberg and Bowman [2004] add *provider-consumer inter-*

faces between successive levels in the *Decide On Action* part, however, not on the *Create Situation Awareness* part, where we need them. However, between the Joint Directories of Laboratories (JDL) levels on *Create Situation Awareness* Kester [2010] does add *provider-consumer interfaces* between one or more consumers and providers.

These interfaces enable goal-directed (top-down) behavior in the abstraction hierarchy on the one hand, where run-time requests are posed by consumers to lower-level agents, and data-driven (bottom-up) behavior on the other hand as providers that *deliver* information according to the requests. Each level l agent requests level $l-1$ information and processes this into *information abstraction* level l information. Not only can provider-consumer interfaces be present between hierarchical components, but also between other sets of components, like the CS and an agent.

Next to the *Core* functionality or algorithm of an agent that processes incoming information—such as a tracking or recognition algorithm, we add another functionality that holds evaluation methods. This functionality is the *Shell*, and surrounds the *Core*, see Fig. 2.1. In this figure it is only displayed for the level 1 agent since our DSS mainly operates through object assessment agents. However, this is not limited to level 1, and can be implemented on each level. The evaluation methods presented in this thesis are RCBE or ATF, but the *Shell* can hold any other method that adapts the service of the *agent* to the current *information requests*, the current awareness (*i.e.* ISA) and the received information from the lower level. Strictly speaking, following Russel and Norvig [2003] these agents are *utility-based* agents because we use utility functions to define how desirable certain features of information are; how "happy" does the agent become of certain features of information.

The combination of *information abstraction* levels and *provider-consumer* interfaces between them enables goal-directed behavior. This starts at the top of the *information abstraction* hierarchy and cascades and splits downwards. For example, multiple ships are on a mission and need to watch out for dangerous situations. The *Impact Assessment* level needs, to assess the danger of the situation, updates of the situation of objects in the environment provided by the *Situation Assessment* level. The *Situation Assessment* level demands the *Object Assessment* level to deliver certain information about objects in the environment to make relations between the objects. Next, the *Object Assessment* level asks the *Signal Assessment* level for detections. Last, the *Signal Assessment* level processes raw data from the collectors into detections. In other words, every abstraction level serves as a provider of certain information to a higher abstraction level.

Interaction Refinement Phase (IRP)

When the interfaces have been set, the interactions can be refined. Throughout the IRP, the higher level $l+1$ agents define the exact contents of the information-requests for the mid-level l team of agents. Information is delivered by the lower level $l-1$ agents—that is from the sensors in JDL level 0—and processed into level l information. The team builds shared awareness from all agents in the team and, ideally, from all collected information.

However, this is often not possible due to constraints of communication between entities. Moreover, even when it is possible, transmission of information may be delayed too long. Therefore, the novel *Shell* functions at run-time as a mechanism that can adapt to dynamically changing information-requests and dynamically changing communication/processing or memory constraints. It simultaneously uses the *information-requests* as well as estimations of the communication situation to enable relevant decision making about *when* to communicate *what* to *whom*.

For this decision making, the interaction with the CS is essential. At run-time, the CS is responsible for the actual transmission of information to other entities and can on request provide accurate and up-to-date communication information—expected consumption of communication resources and expected latency—to the agents. The evaluation methods comprising the *Shell*, in turn, output control information to the CS.

2.1.4 The Distributed Sensor System

The essential features of our DSS are shown in Fig. 2.1 and the properties and behaviors that are determined at design-time will be distinguished from those that are determined at run-time. We propose a system that at design-time is decomposed into multiple entities—in our case ships—and multiple information abstraction levels. From here-on, we make, just like in Albus and Barbera [2005], Steinberg and Bowman [2004], the assumption that the information abstraction levels coincide with the space-time abstraction levels: lower levels of the information abstraction hierarchy of a *Distributed Sensor System* usually have a shorter time horizon and a smaller spatial region for making decisions than the higher levels.

The leading example focusses on constructing an ISA of the tracks of objects, hence centers on the *Object Assessment* level of information abstraction and its interaction with the higher *Situation Assessment* level, the lower *Signal Assessment* level and the CS. The *Object Assessment* agents consist of a *Core* and a *Shell*. During the Interface Configuration Phase, at design-time, provider-consumer interfaces are set between these components.

In this thesis multiple *Object Assessment*—OA—agents form a team that shares a certain awareness of objects. The agents in a team interact by exchanging locally collected information for processing. To interact between entities the CS is responsible for communication and through it the local OA agents engage interfaces with other entities. Communication between components on an entity are assumed to have no constraints.

When the interfaces have been set the IRP starts and in our case transpires all at run-time. Throughout the IRP, the level 2 agents define the exact contents of the information-requests for the level 1 team of agents, for example to which is to maintain and deliver an accurate and timely ISA of the tracks of objects in the environment. Detections from objects are delivered by lower level agents—that is from the sensors in JDL level 0—and processed into tracks by the *Core* algorithm.

The surrounding *Shell* can be implemented with any evaluation algorithm aimed at making run-time decisions that are adaptive to requests and communication constraints. The decision making process involves the following aspects:

1. which level of information to communicate—for example level 1 track or level 0 detections,
2. whom to share information with—for instance refrain from sharing information about a certain object with an entity that cannot have a significant effect on the object, both in action and in observation.
3. what and when to communicate to adjacent agents—such as not sharing these detections of an object that do not decrease accuracy of a certain track versus sharing detections that do,
4. what and when to deliver requested updated state information from a lower level agent to a higher level agent on another entity—based on the current information request.
5. what processing to do—it can be that at one point, recognizing the object is more important than tracking the object, resulting in preference for detections that increase the probability of recognition instead of detection that increase the position accuracy.

We focus on the second and third aspect. We present:

Request and Constraint Based Evaluation run-time evaluates which detections to share with the other team-members by comparing the value for the ISA with its costs for communication. When the value is higher than the costs the detections are communicated, otherwise they are not (Chapter 5).

Adaptive Team Formation run-time evaluates the contribution of entities in the construction of ISA and can exclude and include entities from this construction (Chapter 6)

ATF run-time adapts the interfaces between agents on separate entities at the same level, and the interfaces between agents on successive levels as well as locally between agents. The Communication Service (CS) is set at design-time.

2.2 Identical shared awareness

The phrase ”The problem with today’s systems is not a lack of information, but finding what is needed when it is needed” by Endsley [2010], gives voice to the problem of creating timely *shared awareness* in a DSS generating an explosion of information. Our aim is as well to find what is needed when it is needed by meeting the varying goals that are set for the *shared awareness* through separating the valuable from the less valuable information.

Through the whole extent of the thesis, each team maintains an awareness that consists of shared information and ensures that the awareness remains the same for all entities. Furthermore, the entities know that everybody in the team has the same awareness. Finally, the team has a shared understanding of interpreting and acting on shared awareness. This explains why each entity in Fig. 2.1 has the exact same decomposition. This level of inter-operability is described by Dorion and Boury-Brisset [1998] in the Levels of Information Systems Inter-operability model as the highest level of inter-operability—L4 in the following list of five levels:

L0 no interaction or unintelligible information,

L1 unstructured representation of information,

L2 a common representation of information (syntactic level),

L3 a common understanding of information (semantic level),

L4 common understanding of how information is used (pragmatic level).

It enables the distributed entities to coordinate their behavior. Such an ISA is also described by Kingston and Martell [2004] as comprehensive shared awareness:

> The ideal of Shared Awareness is when entities have Shared Information and are able to act on it. This means that they have internalized (a cognitive function) the information and are aware not only of the information held by the participants, but have some idea of how they will act on it. [Kingston and Martell, 2004, p. 5]

In our case we will define it as a state $\hat{x}$ representing the perceived object o, that firstly, is *identical* and secondly is *synchronized*. The awareness can only be identical if each agent incorporates the same information, wherefor it needs to share its relevant information with the team-members, and all agents need to have common methods to process that information in equal awareness. To keep it synchronized the agents must first have confirmation that the information is received by all and then *simultaneously* update the ISA with that information.

One possible solution for maintaining an ISA is to have all entities communicate all their local observations to all other entities, which subsequently use this to construct the awareness (Mutambara [1998]). Whereas ISA is achieved, this solution does not take communication constraints in to account, which can cause severe delays of communication.

In order to reduce the communication load in Durant-Whyte et al. [1990] and Sawo et al. [2008] a method, Distributed State Estimation (DSE), is described that only incorporates local estimates of adjacent entities. Although their aim was not to achieve ISA, this, and using the Covariance Bounds method to prevent the need for storing information about correlations between entities, greatly reduced the communication load and improved the stability and accuracy of local estimates. DSE successfully reduces communication load, but is not adaptive to dynamically varying communication loads that can occur in a DSS.

Another method that aims to reduce the communication load is introduced in de Waard [2008]. In this article, a method was introduced that fuses single measurements into a smaller sized super measurement to relieve the communication network. Adaptively allocating communication and/or processing to entities is another element of adaptivity known as *resource management*. It allocates resources to entities by their importance for communicating and/or processing information.

Shared but not identical awareness between agents on separated entities in a communication constrained environment is achieved by the several policies presented in Velagapudi et al. [2007]. Similarly as RCBE, when an agent receives a detection—*sensor readings* in their article—it looks at its *value* to determine whether to share it. In our case, the detection would be shared within the team currently maintaining an ISA of the object. In their case however, the value is calculated based on its local goals and its local awareness. Therefore each agent can estimate a different value for the same detection. If the value is high enough the detection is randomly forwarded to another agent, who subsequently determines the value of the detection to determine whether to share it. This process continues until there is an agent that does not find the detection valuable anymore. Hence, this is a method that can distinguish these detections that are important for the team from those that are not. It is an effective method for sharing data when the agents do not know anything about each others goals, awareness and communication capabilities.

2.3 Modeling Communication

The evaluation methods are aimed to optimize the ISA according to the current *information-request* and the current *communication capabilities*. The communication capabilities must therefore be run-time estimated. This can be done by collecting information regarding the used communication technique, environmental conditions, and system properties that influence communication. We require a communication model that includes this information to describe the most important performance indicators of wireless communication systems, such as link stability, throughput and latency, but does not use complex channel models, to be generic for varying communication techniques in a wide variety of scenarios.

Within the scope of communication models, Shannon [1948] designed the most influential one. Fig. 2.3 represents the basic concepts of communication:

- A transmitter that operates on the message to create a signal which can be sent through a channel,
- A channel, which is the medium over which the signal, carrying the information that composes the message, is sent,
- A receiver, which transforms the signal back into the message intended for delivery,
- A destination, which can be a person or a machine, for whom or which the message is intended.

The goal of Shannon was to provide the most efficient way of communication. He developed mathematical models of communication using entropy, taking into account redundancy and noise that can distort the message transmission (comparable to link-stability) and introduced formulas for calculating the capacity of a channel (related to throughput).

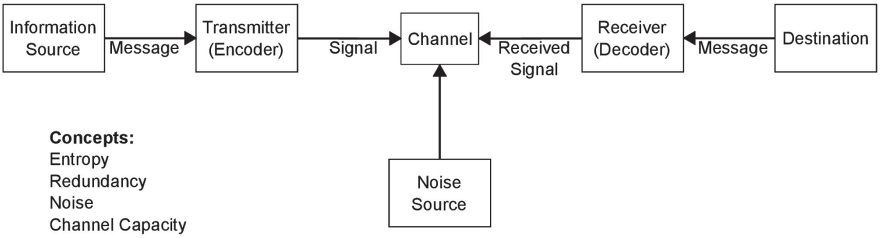

Figure 2.3: The Shannon-Weaver Mathematical Communication Model from Shannon [1948].

Looking at more contemporary research the existing models are too detailed for our purposes. A detailed account on simulating every aspect of the communication between entities, that is simulating the waveforms or signals that flow through the system like distortion effects, noise and interference, is presented in the book Jeruchim et al. [2000]. This book is intended to assist designers in the modeling, analyzing and designing of communication systems up to the detail of complex channel models. It gives detailed analysis on estimating the Signal-to-Noise-Ratio—SNR, which is the starting point of our calculations of the bit-error probability and eventually the message-error probability. While this book presents models up to and including SNR calculations, our modeling requires an abstraction step higher.

Another example of a detailed model is the model in Bianchi [2000] that is based on the IEEE 802.11 standard protocol by Brenner [1997] for communication in a Wireless LAN—*i.e.* Wi-Fi. This protocol consists of a Medium Access Control (MAC) and a Physical layer. In Bianchi [2000] a simple though highly accurate analytical model of the throughput when using the 802.11 Distributed Coordination Function—DCF—is presented. In Vu and Sakurai [2006] also a detailed analytical model is provided for the standard model but then in determining the probability that two or more entities are transmitting at the same time (collision probability). In van Foeken and Kester [2009] we also developed an initial model of the IEEE 802.11 standard protocol. The problem for us is that these models are only valid for the IEEE 802.11 standard protocol and are too detailed to be applied for other and more advanced communication techniques.

While the standard model does include some features that are required for our model other features are missing. Included are fast, dynamic, P2MP and long distance communication but techniques such as WiMAX, Bro@dnet and LARA improve on all these factors. Moreover, they improve on link-stability, throughput as well as latency, provide more functionality for P2MP communication and more adaptivity in resource management:

WiMAX World-wide interoperability for Microwave access (WiMAX) is a fully TCP/IP-based terrestrial wireless communication technology that accommodates faster data rates and longer transmission ranges (up to 50 km) than for example Wi-Fi. In addition, WiMAX features point-to-multipoint—P2MP—communication.

BRO@DNET (TELEFUNKEN RACOMS) Tel [2008] is a broadband wireless military backbone system that uses WiMAX and further provides high flexibility and scalability, and is aimed for military applications.

Layered Architecture for Real-time Applications (LARA) Thebault [2007] is a project that aims to improve the network in real-time distributed applications in a joint (between countries) military context. It resulted in a technology which supports resource management (*i.e.* it actually provides resource management based on requests from Providers), real-time communication, P2MP to produce a Common Awareness and Provider-Consumer interactions.

From the demonstrator that resulted from the LARA project it was shown that latency can be bounded if large amounts of resources (bandwidth) are reserved in the network Systems [2008]. It was also shown that the quality of service in an IP network is not only a matter of bandwidth, but latency is probably as important and more demanding.

These improvements will likely lead to more frequent application of these newer technologies in the DSS of the future. For this thesis we take WiMAX as an example, since this is the most well-known technique of the three with much information on its specifications. We point out that we are not looking for a comparison between the previously mentioned techniques. The most important improvements of WiMAX compared to Wi-Fi regarding our leading example are the longer distance—50 km versus 95 m—and the use of channel bandwidths of 25 MHz to provide data-rates up to 134 Mbps compared to a maximum of 54 Mbps for Wi-Fi. Also in comparison with Link 16—a widely used generic tactical datalink system—WiMAX provides higher data-rates. Although Link 16 provides long-distance communication, the mere maximum of 1Mbps does not compete with the 134 Mbps.

In order to simulate technologies such as WiMAX we needed to improve the model presented in van Foeken and Kester [2009] and therefore presented an innovated model in van Foeken and Kwakkernaat [2011].

Next to being able to model these technologies the evaluation methods rely on accurate and up-to-date communication status information—the expected delay and expected cost of communication. Several models exist that estimate the expected delay in communication within multi-entity networks. For example, Drozdowski [2002], introduce a model that estimates the total execution time of parallel applications—*i.e.* applications that run on multiple entities and require communication. They do this by including estimation of communication delay. However, this estimation is done off-line where we want do this online. Moreover, an unloaded deterministic network is assumed and can therefore not account for dynamic influences on the delay—such as moving entities, changing resource allocation.

Ozdemir et al. [1999] present a method of estimating the delay of sending a message to multiple receivers—*i.e.* multicast—by transmitting low-overhead messages over the network and measuring the delay. By doing so an estimation of delay is made. This is a method of delay estimation without making use of knowledge of certain variables of the system architecture or the environment. In order to estimate the delay we aim to take many variables into account, such as bandwidth, system losses or atmospheric conditions.

2.4 Utility and Value of Information

2.4.1 Utility

To our knowledge the authors of Eswaran et al. [2010, 2011] are among the first that realize that *utility* and *value* in networked systems should be about the usefulness of information in relation to the system-goal. They give a thorough account of several interpretations of utility in literature, but

> although all the related work provides different interpretations of utility, none of them succeeds in providing a subjective, user- or task-based assessment of the value[1] of the data, which is the true purpose of using utility-based adaptations. [Eswaran et al., 2011, p. 2]

Some work defines mathematical utility functions of communication constraints such as bandwidth and latency (Kelly et al. [1998], Low and Lapsley [1999], Lee and Chiang [2006], Jin et al. [2005]). Other work takes distortion or signal-to-noise ratio as a direct utility measure (Chou and Zhourong [2006], Setton and Girod [2007]). Another way of measuring utility of data is with information-theoretic entropy-based methods, mainly applied to sensor-management (Grocholsky [2002], Wang et al. [2004], Aoki et al. [2011b], Kreucher et al. [2003, 2005a,b], Bolderheij et al. [2005], Gulrez and Kavakli [2007]).

The problem with these approaches is that they are [Eswaran et al., 2011, page 2]:

> insufficient and often lack practical relevance. In reality, the utility of data depends on a diverse set of factors, such as the goal of the receiver, how the data is processed by the receiver, how valuable the data is for the receiver to achieve his goal, etc. Understanding information utility, along with defining it in an application- and type-agnostic way, and developing methods to quantify it, is an open and important area of research.

Eswaran et al. [2011] realize that it is important to measure the contribution of data to the system-goal, especially in mission-oriented networks. They acknowledge that only measuring the quality of the network is not enough, nor solely measuring the quality of the data, but that a more direct link is needed with the mission objective. They have developed a metric that characterizes *utility* of data in terms of how much it impacts the *accuracy* of the task and the *timeliness* of

[1] Eswaran et al. [2011] do not define value explicitly and therefore does not hold the same meaning as in this thesis

task-completion. Similarly, we optimize the *identical shared awareness* through *information-request*s, formalized by *utility function*s, that describe the conditions on the *accuracy* that data should bring and the *timeliness* of synchronization.

The difference is that they directly measure the utility of data for the mission-objective, whereas our system consists of multiple parts with possibly multiple sub-objectives. The *information-request* is such a sub-objective and the evaluation-method calculates the *utility* of data given this *information-request*. Nevertheless, the philosophy of aiming to optimize the mission objective is the same for both approaches. The first step of our method is the derivation of a *utility function* from the current mission objective towards the objects in the visible environment. The second step is the evaluation of the impact the data on the *identical shared awareness* using this *utility function*.

As said by Eswaran, utility depends on several factors:

- goal of the receiver—*information-request*
- how the data is processed by the receiver—what processing method is used in the core of the *object-assessment* agent to process the data
- how valuable the data is for the receiver to achieve his goal—how much gain in utility has the data for the important features of information

One technology that optimizes behavior of a DSS by *information-requests* is *Sensor management*. *Sensor management* is a technology that controls the physical allocation and actions of sensors. The papers Grocholsky [2002], Kreucher et al. [2003, 2005a,b], Bolderheij et al. [2005], Gulrez and Kavakli [2007] present sensor management methods that select the best sensor action based on information gain or *information divergence*—which is an information-theoretical measure.

Information divergence—such as Kullback Leibler-divergence or Alpha-divergence (Zhou and Chellappa [2006])—is a measure of difference between two probability distributions. In Velagapudi et al. [2007] and van Foeken and Kester [2009], functions are presented that both calculate the *expected value* in terms of gain in *utility*[2]. This gain is the difference between two information divergences: the information divergence between the posterior estimate and the reference state, and the prior estimate and the reference state. Although it works for comparing two probability density functions, the problem is that information divergence is an implicit and indirect measure of comparing the utility of two states. In chapter 4, we introduce a new way of calculating the utility that overcomes the implicitness of information divergence.

2.4.2 Value

Utility and *value* are terms with many definitions. We define them as follows:

Utility, $\mathcal{U}$, is the usefulness of the important features of information.

[2]Although the principle is the same it must be noted that in Velagapudi et al. [2007], instead of a gain in utility, a reduction of cost is applied

Value, $\mathcal{V}$, is the utility that is gained by newly gathered data.

In other words, utility is the usefulness of a certain state of information, and value the added utility or impact of data on the current state of information.

In this thesis, *expected value* $\hat{\mathcal{V}}$ is defined as the gain in utility:

$$\hat{\mathcal{V}}(\Gamma, \mathcal{Z}) = \mathcal{U}\left(\Gamma\left(\hat{X}|\mathcal{Z}\right)\right) - \mathcal{U}\left(\Gamma\left(\hat{X}\right)\right), \tag{2.1}$$

where Γ are the important features of information, $\hat{X}$ the state estimate and $\mathcal{Z}$ the data. In other words, how much has the posterior—state estimate $\hat{X}$ with data $\mathcal{Z}$—improved over the prior state estimate. If, for example, the prior state estimate has the same utility as the posterior state estimate there is no gain in utility so no value. The *expected value* is calculated by a *value function* and was inspired by the work presented in Velagapudi et al. [2007]. In their article they present policies which are able to maintain a shared awareness of the state of the environment in a communication constrained multi-component team, by locally balancing information *value* against communication *cost*. The policies are based on the assumption that components do not know anything about one another. Information gain is locally determined and the relevant global information is assumed to emerge from the local interactions. Their value function measures the *expected value* of detections and several policies use it to decide, although in different ways, to transmit or not.

Their value function is a vast improvement over approaches applying *information-driven* sensor management. Information-driven sensor management strives to select the action that will result in a measurement that influences the state such that it maximizes the information gain (or minimize the uncertainty), mostly between the posterior and the prior estimate. The first improvement of the value function is that it attempts to find the improvement of the posterior over the prior *not* by simply taking the information difference between the two, but by comparing both with the actual state[3], like with task-driven sensor management. By comparing with the actual state the measure becomes absolute instead of relative. The second improvement is that a cost function is used to represent the subjective importance of the divergence.

We adopt this function with some changes. The first change is actually mostly a matter of taste and does not change the concept. Where Velagapudi et al. [2007] signify the *cost* of information difference we preferred reasoning with a more *positive* term, *utility*, which determines the usefulness of information. The second change is that we use the *integrated utility function* to calculate *utility* instead of a function that uses information divergence.

2.5 Utility and Cost Evaluation

What we are aiming at is developing methods that are adaptive to dynamic *information-requests* and dynamic *communication capabilities*. The first evaluation method in this thesis is RCBE. RCBE decides to communicate detections

[3]From their article it is unclear what the actual state represents, but we expect that it is the best possible estimate by taking in all available information associated to the state.

or not, based on their *expected reward* by subtracting the *expected cost* from the *expected value* of communication: $\hat{\mathcal{R}} = \hat{\mathcal{V}} - \hat{\mathcal{C}}$. A positive *expected reward* results in communication and a negative one for disposal. What are other techniques that do simultaneous value and cost evaluation, what can we learn from them and what are the aspects we need that are not present? We discuss Ming et al. [2007], Velagapudi et al. [2007], Marck et al. [2008], Eswaran et al. [2011].

Besides these, there exist fusion methods that ensure a lower size of the information by aggregating measurements as well as a high quality of the information. Fusion methods are useful for balancing quality and size of data but do not account for the real-time resource availability and do not base decisions on dynamically changing requests but on fixed information-theoretic measures. Information-theoretic approaches indeed balance the value of information with the cost of communication, but the measures for valuing are fixed, hence not adaptive to the dynamically changing goals. Then there is *resource management*, which is a method that optimizes according to resource-requests but is not concerned with how these requests are made.

In the previous section we mentioned Eswaran et al. [2011] and they realize that both the quality of data and the quality of the network needs to be taken into account in measuring the contribution of data to the system-goal, especially in mission-oriented networks. Informative is their example that represents a mission that is dependent of timely and accurate track information of enemy entities. They show that the accuracy and timeliness of track location information degrades with increasing packet-loss. This exactly points to the problem of balancing *value* and *cost* of information. If more resources are invested in transmitting information, packet-loss is decreased and value is increased. Balancing cost and value is exactly what the evaluation methods in this thesis aim to do.

Another technology that adapts communication to the relevancy of information is presented in Ming et al. [2007]. It evaluates the relevancy of information and then orders the data compression and transmission of competing pieces of information based on this relevancy. This technology does, at run-time, perform information-request evaluation and uses this to determine the order of compression and transmission but does not adapt to the run-time communication capabilities.

A control mechanism is described in Marck et al. [2008] for efficient communication between entities providing state estimation of objects and a decision support system assessing the situation and determining the impact of certain situations. Efficient communication is reached when the entities only communicate state information to the decision support system if the state has sufficiently gained in information according to the requested information by the decision support system. In other words only relevant information is communicated. The concept in Marck et al. [2008] is aimed for relevant communication between *different* levels of information abstraction. It can be seen as the 4th type of decision making in section 2.1.4. In contrast, we aim at relevant communication between spatially distributed components of the *same* (*object assessment*) abstraction level. The entities providing state estimation of objects act on the *object assessment* abstraction level and the decision support system on the *situation assessment* abstraction level. Another difference is that in Marck et al. [2008] the measure for relevancy is a fixed information-theoretic measure.

To return to the method presented in Velagapudi et al. [2007], their method contrasts with our method as follows: We assume all entities have identical methods, procedures and algorithms in order to process information equivalently (*i.e.* commonality), where they assume global information emerges from local interactions. Commonality has the advantage that individual entities may directly determine the global *expected reward* of local information. Moreover, where shared awareness is not identical in Velagapudi et al., our issue is to maintain an *identical shared awareness*. This has the advantage of enabling coordinated distributed action. Lastly, in Velagapudi et al. [2007] there is no account for the effects of latency of communication.

The problem of creating *identical shared awareness* by a *Distributed Sensor System* in dynamically changing environments was first tackled in van Iersel et al. [2008]. The architecture presented by Kester [2008] was followed to design systems with provider-consumer interactions that enforce data evaluation guided by *information-request*s. A simple data evaluation algorithm, the *association method*, was used to determine transmission of detections or not. It was promising that this simple method excluded detections without compromising the quality of the *identical shared awareness* and therefore relieved the communication channel. The *Request and Constraint Based Evaluation* method was developed after and also inspired by this work.

2.6 Identical Shared Awareness in Adaptive Teams

The second method that simultaneously evaluates value and cost of information is Adaptive Team Formation.

Bolderheij et al. [2005], introduce a sensor management approach that, in case of limited resources of processing or communication, performs a priority assignment per *object*. This assignment is done by evaluating the risk that each object brings to the completion of the current mission objective. Thereafter, a sensing function is assigned to an object, like a tracking function to do on a certain object in the detection range. Following a sensor is selected to perform the task. Then, in case multiple tasks are assigned to a sensor, the sensor performs a resource allocation based on the priority of the tasks. This method inspired some of the features of our ATF. One of them being to evaluate the contribution of sensors to single objects instead of a group of objects, since one object may endanger the mission in a totally different way than the other. One can imagine that different features of information are important when dealing with a missile threatening an entity compared with a Naval mine. Where Bolderheij et al. [2005] selects a single sensor to do a task, we show the advantages of multiple sensors observing objects. Moreover, ATF determines at run-time the expected reward that each sensing entity can bring to the state estimation of certain features of the object. In this way teams are dynamically organized per object.

In order to cope with limited processing and communication resources in networks of cameras that perform automatic localization and tracking of people, Spaan and Lima [2009] apply dynamic sensor selection based on user-defined objectives. They use Partially Observable Markov Decision Processes (POMDPs) to decide which set of cameras is active in the next time-frame. POMDPs assign rewards to actions and its resulting states. The reward is based on the number of cameras used, which defines the costs of resource use, and the value in satisfying either the coverage objective or the uncertainty objective or both. Where the coverage objective is satisfied when the persons to be tracked are observed by at least one camera and the reward in terms of uncertainty increases with lesser uncertainty. Even though they focus on one specific application their method is highly generic. Generic in the sense that it can be applied to other applications with different sensors, different tasks and different user-objectives. But also generic in how to define the costs of resource use. We believe we can install ATF in this generic method. However, our method is novel, firstly, because it particularly focusses on constructing ISA, and secondly, because it uses a new mechanism of determining utility and reward of information.

In achieving an optimal ISA of an object in case of limited communication, work has been offered that forms coalitions of sensors that share information about the object, based on the quality that the sensors bring and the communication limitations. Howard and D. Payton [2002] present such an approach, by forming a coalition that shares all measurements and occasionally updating non-members with tracklets (an aggregation of multiple sensor successive measurements), and thereby using less bandwidth. Through this method they achieve ISA—they call it the *identical tactical picture*—between members and a delayed ISA between non-members. This delay is caused by the time it takes to form tracklets. They suggest three methods to determine the coalition members. All three methods achieve dynamically changing teams. At the base of these methods stands a certain information theoretic measure to find the reduction of uncertainty by measurements to the track estimate. A non-member could then for example join if the reduction exceeds a certain threshold. The first method tracks the values of this reduction over time. When two entities are competing for membership and have the same reduction at some point in time, the one with an increasing contribution should be preferred over one with a decreasing contribution. The second method aims to achieve small coalitions with optimal—that is with many different viewing angles—positioning with respect to the object. The third method aims for coalitions with members with varying qualities.

These three methods present useful approaches to coalition forming. Compared to ATF there are a few differences and a few overlaps. One main difference is that they either share all measurements or share tracklets, while in case of coalition we also use RCBE as an evaluation method for single measurements and in case of no coalition we do not share anything. The advantage of using RCBE is that it respects the current communication capabilities, while their method will have trouble in cases where communication capabilities are to low to cope with high amounts of measurements. Sharing tracklets or some state once in a while might be a valuable addition to our ATF. Another difference is that their method is focussed on tracking alone, while our ATF is more generic. What is promising of their first method is that they compare the contribution of entities over time and compare the track-records with each other. ATF also locally keeps track of the contribution of an entity and uses this to make decisions about excluding entities from the team. They use an information-theoretical measure to find the reduction of uncertainty. In section 2.4.1 we have claimed that such a measure is too indirect and implicit and will explain this further in chapter 4.

2.7 Performance

The performance of our evaluation methods needs to be determined. Usually performance is reported in a way that facilitates comparisons of experimental results with other work. An example of such a performance measure is the Mean Squared Error (MSE). The goal of a system would be to optimize the MSE of some estimated state and compare it with other work. In our systems, however, goals are reflected in *information-requests* with *subjective* demands for certain important features of information. The goal is therefore not set in stone and, moreover, unique, making it hard to compare results with other work.

In chapter 5 MSE is used to compare the performance of our method with a benchmark method, but there it was possible because the point of the experiments was to show that higher accuracy is achieved with our method due to evaluating the contribution of data to the track estimate.

In chapter 6, however, performance measures are used that relate directly to the current *information-requests*. In the experiments we use *reward = value - cost* to determine the performance of the method. Value is directly related to the information request, and costs to the amount of communication resources used. Combined it reflects the performance of the method. The higher the reward the better the method is. In run-time decisions are made by estimating the *expected reward* of one action over the *expected reward* of another. By comparing the actual reward with the expected reward we can determine the estimation quality of the reward. And by comparing the actual reward in multiple settings we can find the best setting.

2.8 Experimentation Environment

For the simulation environment for our experiments the system of van Iersel et al. [2008] was used. The environment is shown in Fig. 2.4. It consists of four federates which communicate and synchronize through a Run Time Infrastructure (RTI) developed by Nederlandse Organisatie voor Toegepast Natuurwetenschappelijk Onderzoek (TNO) (Jansen et al. [Spring 2004]). Communication with the RTI runs through the Run Time Communication Infrastructure (RCI). The federates that we used were:

- The Joint Research On Air Defense Simulation (JROADS)) takes care of visualization of the scenario as well as running the scenario.
- The sensor federate keeps track of all the sensors, such as the radars on the ships, and does the production of contacts.

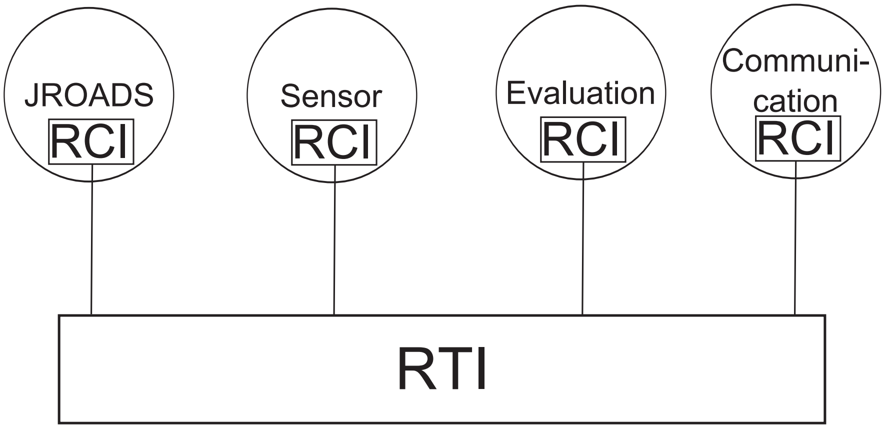

Figure 2.4: The schematic layout of the software demonstrator

- The communication federate is implemented with the functions of the CS, that is, communication between entities and estimating the Expected Delay Distribution (EDD) and Expected Cost of Communication (ECC).
- The evaluation federate harbors the evaluation methods RCBE and ATF and performs tracking of objects and the evaluation of detections.

Our leading example consists of a multiplicity of ships maintaining an ISA of the objects in the environment. The structure of the experimentation environment is somewhat counterintuitive since it is *not* divided into separate entities, each running their own sensors, tracking and evaluation. Instead, the structure was motivated from a perspective of functionalities; hence is divided in a sensing, communication, evaluation and visualization functionality. These functionalities can interact, as said through a RTI. Fig. 2.5 shows the parts of the system decomposition of Fig. 2.1 that each federate performs.

The realism of this demonstrator lies in the realistic simulation of the radars, the creation of contacts, the environmental features that are imbedded in JROADS, and the rather detailed communication model. Moreover, the RTI supports High Level Architecture—HLA—and IEEE 1516 standard.

2.9 The system

This section formally introduces some of the variables used in this thesis. $\mathbb{R}$ and $\mathbb{R}+$ define the set of real numbers and non-negative numbers. Our DSS consists of entities $\mathcal{J} = j_1, j_2, j_3, ...$ These entities incorporate radars that produce detections of object positions, $\mathcal{Z} = [z_1, z_2, z_3, ...]$, out of an uncertain environment $X \equiv \mathbb{R}$. The radars surveil the environment for visible objects $\mathcal{O} = [o_1, o_2, o_3, ...]$ Referring to Fig. 2.1, each entity j has multiple information abstraction levels, where the

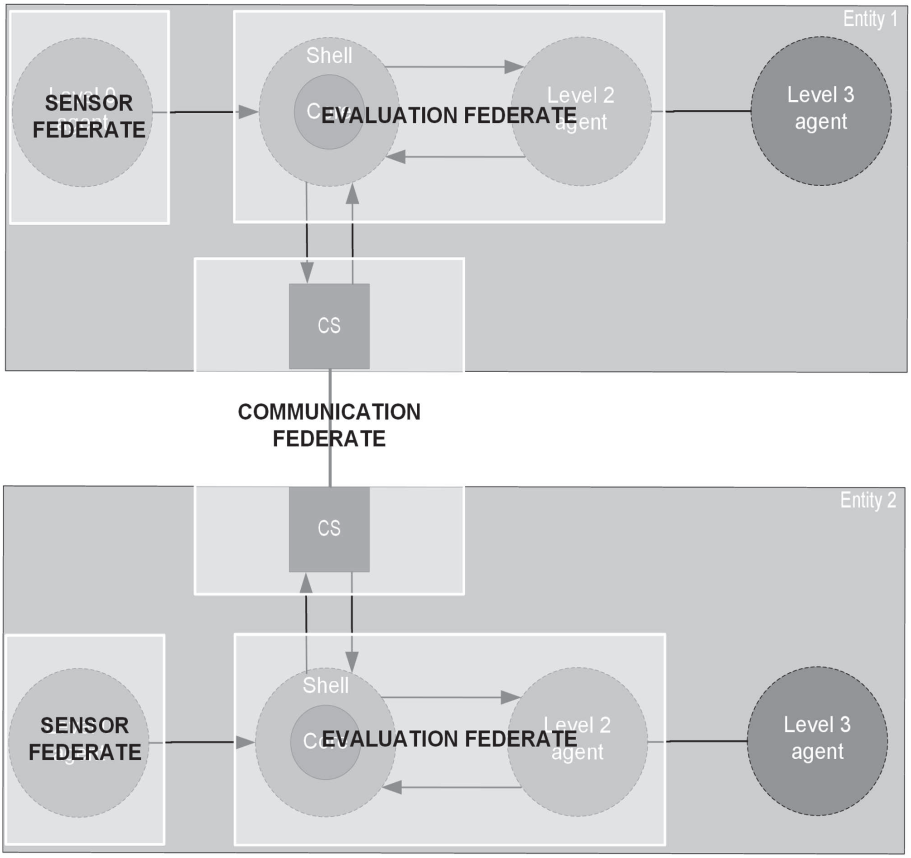

Figure 2.5: A mapping of the experimentation environment over the system decomposition.

evaluation methods are acting on the *object assessment* level 1 alone. Each entity can harbor multiple agents $\mathcal{A}_j = [a_{1j}, a_{oj}...]$. RCBE and ATF are part of an agent, where an agent is defined as an autonomous component acting on an entity, j.

Each agent is cooperating in a team with other agents, $\mathcal{S}_o$, to observe and act upon a singular perceived object o—$o \in \mathcal{O}$. This cooperation consists of maintaining an *identical shared awareness* (ISA), $\hat{x}_o$, of the object. All ISAs of all visible objects are $\hat{X} = [\hat{x}_1, \hat{x}_2, \hat{x}_3, ...]$ Each agent uses a *core* functionality that is a domain-dependent algorithm ϕ fusing level 0 detections, $\mathcal{Z}_o = [z_1, z_2, z_3, ...]$, that associate to object o, into level 1 updated state estimates, $(\hat{x}_o \mid \mathcal{Z}_o) = \phi(\hat{x}_o, \mathcal{Z}_o)$. Usually there is not much ambiguity in the track-to-object association and therefore we assume that each ISA, $\hat{x}_o$, relates to a single perceived object o. Therefore it is also sufficient to have a single-track-to-object tracker without track-to-object ambiguity. The choice for the filter bears no significance towards the functionality

Table 2.1: Links between agents and entities

entity/agent	a_{11}	a_{21}	a_{31}	Total
j_2	0	1	1	2
j_3	0	1	1	2
j_4	1	0	1	2

of RCBE, and in the experiments we use the Kalman filter (Kalman [1960]).

Important to note is that an agent, a_{oj}, comes into existence when the entity j starts participating in the team of object o. In practice an agent is born when the entity finds a new track or when the entity contributes positively to a track maintained by other agents than itself and is allowed to partake in the team.

The agent also has a *Shell* that can contain the two adaptive functions of this thesis: Request and Constraint Based Evaluation (RCBE) and Adaptive Team Formation (ATF). ATF is a function defined as $\varphi(\mathcal{S}_o, \hat{x}_o, \mathcal{Z}_o)$, which determines whether the agent should be in team $\mathcal{S}_o$, based on the contribution the information $\mathcal{Z}_o$ brings to the shared state estimate $\hat{x}_o$. RCBE is a function defined as $\psi(\mathcal{S}_o, \hat{x}_o, z_o)$, and determines whether single observations, such as z_o, should be shared with team $\mathcal{S}_o$ based on its contribution to $\hat{x}_o$ and therefore used to update $\hat{x}_o$: $\hat{x}_o \mid z_o = \phi(\hat{x}_o, z_o)$.

Imagine for example there is a situation with entities $\mathcal{J} = j_1, j_2, j_3, j_4$. There are three perceived objects o_1, o_2, o_3, observed respectively by team $\mathcal{S}_1 = [a_{11}, a_{14}]$, $\mathcal{S}_2 = [a_{21}, a_{22}, a_{23}]$ and $\mathcal{S}_3 = [a_{31}, a_{32}, a_{33}, a_{34}]$. Entity j_1, has therefore three agents, a_{11}, a_{21}, a_{31}, sharing $\hat{x}_1, \hat{x}_2, \hat{x}_3$. Table 2.1 intuitively illustrates how many ISA's entity j_1 is sharing with which entities:

Moreover, it gives insight into the maximum amount of data that entity j_1 would have to transmit and what data needs to be transmitted to which entities, after a detection moment. The example shows that when there are three detections of the three objects respectively, and they are all considered relevant by the RCBE, the CS will have to transmit 2 detections to each entity.

2.10 Conclusions

This chapter covered the state-of-the-art of the main research ideas of this thesis.

To start with we looked at architectures for designing a DSS. We have learned that it can be valuable to distinguish beforehand the properties or behaviors that are determined at *design-time* from those properties and behaviors that are determined at *run-time*. Although researchers observe a trend where systems perform more and more processes at *run-time* and do state which parts are determined *design-time* and *run-time*. However, they do not explicitly separate *run-time* and *design-time* features beforehand.

We can conclude that decomposition of a DSS requires two principles, *information abstraction* and *multiplicity or parallelism due to—physical—constraints*. Several existing architectures use these principles and we have combined features from two similar architectures presented in Steinberg and Bowman [2004], Kester [2010]. In addition, by performing this two-dimensional decomposition 'functional' agents are constructed that act on a certain information abstraction level and on a spatially separated entity. Moreover, to facilitate communication and estimation of communication capabilities we added a Communication Service.

We also found that the state-of-the-art architectures do provide steps to define the interfaces and interactions between components but do not distinguish a phase where the interfaces between components are set (Interaction Configuration Fase) from a phase where the run-time interactions over these interfaces (Interaction Refinement Phase). We introduced such a partition which makes the design process more modular and clear.

Between the levels of information abstraction we required some mechanism that accommodates the requesting of information from the higher abstraction level and the provider of information given the *information-request*. Kester [2010] present a solution in *provider-consumer interfaces*. On top of that we needed some functionality within an agent that could hold the evaluation methods. Therefore, we have introduced the *Shell* that surrounds the *Core*.

After having finalized the architecture, essential properties of ISA were given. From Dorion and Boury-Brisset [1998] the high level of inter-operability that we want to achieve is extracted, and from Kingston and Martell [2004] the ideal of being able coordinate behavior on the ISA. We concluded that ISA should in addition be synchronized.

Then, we focussed on modeling communication. We argued that it is required to be able to estimate the current communication capabilities—expected delay and expected cost—so that the evaluation methods can reliable estimate the *reward* of information. Some approaches to modeling communication were too detailed to be generic for a variety of communication techniques. Other approaches modeled expected delay, but either this was done off-line with a network that was too fixed, or the estimation was done without including parameters. We can conclude that we need a communication model that is realistic, such that estimation of delay and costs can be done online and that it describes the most important performance parameters. We do not, however, want to use too much detail to be generic for varying communication techniques in a wide variety of scenarios.

Our idea of how utility should be measured aligned largely with the work of Eswaran et al. [2011], because they also think utility should be about the usefulness of data to the system-goal. Several other approaches were discussed but concluded that these often lack practical relevance and only relate utility to part of the set of factors that we think are all important—*i.e.* only bandwidth and latency or only information-theoretic entropy-based methods. We required, as did Eswaran et al. [2011], an approach that included *information-requests*, current state, and current communication capabilities.

Then we looked more into the details of the utility function. The utility function should be able to relate the *information-request* directly to the current estimated state. In literature there are some approaches that use information divergence, but we concluded these did not fulfill our need for directness.

To calculate the value we needed a measure that is absolute. For this we learned a lot from Velagapudi et al. [2007]. We adopted their value function with some changes.

Ultimately, we need methods that construct ISA adaptive to *information-requests* as well as the current communication capabilities—*i.e.* the expected delay and costs. We can conclude that measuring value information-theoretically, as is done in fusion methods and several articles Marck et al. [2008], Velagapudi et al. [2007], is too rigid for our purpose of being adaptive to different *information-requests*. We can also conclude that methods applying resource management based on requests do adapt to *information-requests* and take the limited communication capabilities into account. However, we would like to also adapt to them.

With respect to ATF, Bolderheij et al. [2005] inspired us to form teams around objects instead of groups of objects. Two very promising approaches to dynamic team formation are presented in Spaan and Lima [2009], Howard and D. Payton [2002]. In conclusion, they both present methods that, at run-time, adapt the team formation to dynamic goals and dynamic costs, where the last one even aims to construct ISA. The similarities are significant, but we would like to have an ATF that evaluates and deals better with the run-time communication capabilities and uses a more direct and intuitive way of determining utility.

Regarding systems with alternating *information-requests*, we conclude that the performance should have a direct relation with the *reward* of information.

Chapter 3

Modeling Communication

3.1 Introduction

As stated in the introduction and state-of-the-art chapters, the evaluation methods require up-to-date information regarding the communication capabilities at the moment evaluation is needed. Therefore, this chapter introduces a communication model that is able to estimate at run-time the

1. *Expected Delay Distribution* (EDD),
2. *Expected Costs of Communication* (ECC).

The EDD reflects the expected probability of successful communication over time. Latency or delay in communication, caused by constraints in the communication and processing, negatively influences the utility of information, especially in time-critical situations. Determining the real-time utility of information, therefore, requires knowledge about the current expected delay of communicating information.

The ECC reflects the amount of used resources over time, with respect to the total amount of resources available. In this thesis costs are considered the counterbalance of utility, and together make up the reward of information. In case of RCBE, if costs of communication are higher than the gain in utility of that information the reward will be negative, hence the information will not be communicated. Therefore, to evaluate the reward of information estimating the costs of communication is essential.

Modeling a communication technique and estimating the current communication capabilities is based on the most important performance indicators: *link stability, throughput and latency*. To be able to model different communication techniques the model should not be too detailed, but also not too generic to ensure realistic results. The goal is to bridge the gap between statistical models and the physical understanding of the channel without using complex channel models. In the literature it is hard to find suitable generic, low-complex models and therefore such a model is introduced here.

This chapter is largely based on material from [1] and we use WiMAX IEE [2002] as an example communication technique. It is a promising (civil) high data-rate terrestrial communication system that features fully TCP/IP-based terrestrial point-to-point or point-to-multipoint communication that can serve multiple subscribers over distances of up to 50 km when using stationary, line-of-sight connections. Especially the ability of transmitting with high data-rates over long distances makes it a good option for communication between ships on the ocean.

We distinguish three types of parameters that are being used to model the communication:

- parameters that are related to the communication technique—such as bandwidth and data-rate,
- parameters that are related to the environment—such as roughness of the sea,
- parameters that are related to the system—such as system temperature.

Some of these parameters are fixed, such as the bandwidth or environmental conditions, others are tunable. The experiments are two-fold: firstly, the influence of varying environmental conditions on the EDD and ECC is shown, secondly, the influence of tunable parameters on them is shown.

In a realistic implementation it may not be feasible to derive the EDD due to the limited time and delay information that is available at each instant. The EDD may therefore be determined using an analytical model that is tuned using parameters determined from active measurements in the network.

This chapter starts with a description of the communication model and how it functions in interaction with the other components of the DSS. This is followed by presenting the factors that constrain the *link-stability*, *throughput* and *latency* in section 3.3. Section 3.4 presents the derivations of the EDD and the ECC and ends with describing the simulation of communication between entities. This is followed by an example of estimating the EDD and ECC in different environmental circumstances in case the CS uses WiMAX as a communication technique to communicate with 3 other entities (section 3.5). Finally, conclusions are given and future work is discussed in section 3.6.

3.2 Overview

Fig. 3.1 illustrates the functionality of the communication model. A new message arrives that can be either a *local detection* or a *remote detection*. In the latter case the detection simply updates the ISA. In the former case the detection is up for evaluation. On cue of a local detection an evaluation method requests from the *Communication Service*—CS—the current *Expected Delay Distribution*—EDD—and *Expected Cost of Communication*—ECC. They use this to evaluate the local detection. When an evaluation method decides to *transmit a message* it can request

[1](van Foeken and Kwakkernaat [2011])

the CS to do so. We recall that an agent consists of a *Shell* and a *core*. The *Shell* harbors the evaluation methods and interacts with the CS. The evaluation method can add certain conditions to this request, like a maximum delay or a maximum amount of transmission attempts. Subsequently, the communication model will produce a real delay and a real amount of transmission attempts, possibly constrained by the request. Although not shown in the figure, another option for the evaluation method is using *resource management* to allocate more or less resources to the transmission of a message. The last service that the CS can deliver is *re-queuing* a message. In chapter 5, RCBE makes use of this service. As described in the introduction, RCBE performs transmission only if a detection has a positive *expected reward*. In chapter 5 we show that over time, the reward of a detection decreases and that re-evaluation of the detection after a failed transmission attempt can be valuable. Re-queuing is requested if the re-evaluation resulted in a positive *expected reward*.

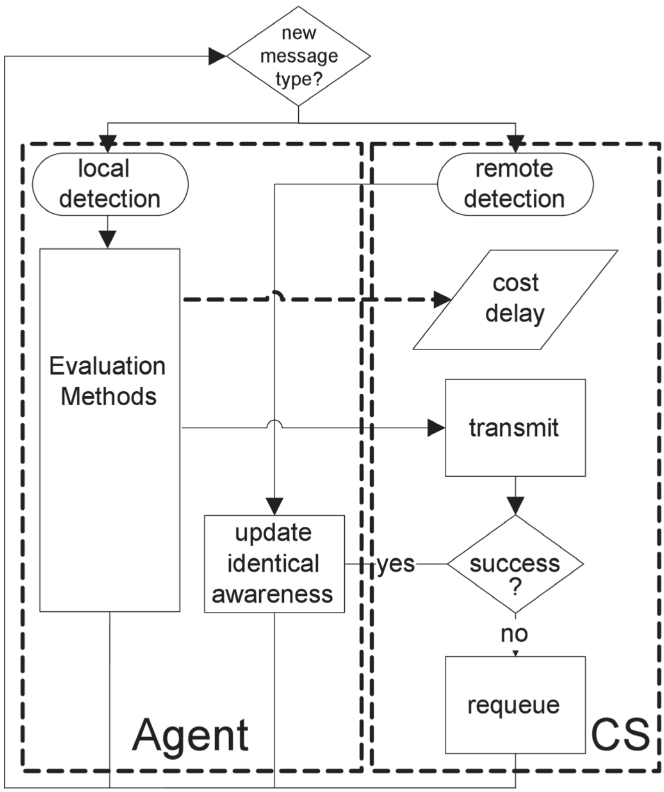

Figure 3.1: Interaction between agent and Communication Service

3.3 Communication Constraints

As said in the introduction, the main performance indicators of wireless data link systems are *link stability*, *throughput* and *latency*. In this section the important underlying parameters for our model that influence these indicators are discussed.

There are parameters related to the system, communication technique and the environment. The parameters related to the system are fixed, but parameters that are related to the environment need to be estimated. The parameters related to the communication technique are partly fixed and partly tunable.

Link-stability and throughput are mainly constrained by the *signal-to-noise-ratio*—SNR—at the receiver. Parameters that we use in our model that influence the SNR in wireless communication systems:

- transmit power [W]: The amount of power of Radio Frequency (RF) that a transmitter is able to produce.
- antenna/array gain (Multiple-Input Multiple-Output (MIMO)/beamforming) [dB]: the power gain that is achieved by using multiple antenna's at both the receiver as the transmitter.
- operating frequency (clock rate) [Hz],
- bandwidth [Hz],
- data-rate [bps],

There are other important parameters that we do not use to model the SNR, because they are either too detailed or are assumed to be fixed in our communication model. One such parameter is the modulation format. Modulation is the process of varying one or more properties of a high-frequency periodic waveform, called the carrier signal. The three key parameters of a periodic waveform are amplitude, phase and frequency. See Hamid et al. [2009] for motivations of choosing the right format, such as robustness against noise and channel impairments. We assume a Binary Phase Shift Keying modulation technique with WiMAX. Channel coding, Wang et al. [2001], is encoding the message in a redundant way by putting an error correcting code in the message. Examples are block, convolution and Reed-Solomon error coding techniques. Coding improves reliability. Spread spectrum methods are methods that spread the signal over the frequency domain thus resulting in a wider bandwidth signal and have as goal to be more resistant to interference, noise and jamming. Lastly, there is equalization (*e.g.* inverse channel filtering) and diversity (*e.g.* space, polarization, frequency, time). Diversity is the technique of using two or more communication channels with different features. It is meant to make communication more robust against fading and co-channel interference. Channel coding, spread spectrum methods and equalization and diversity are all quite specific and detailed techniques, not the most significant parameters for our purpose.

The parameters transmit power, bandwidth (which impacts the data-rate) and operating frequency are parameters that are tunable by resource management methods. These parameters are tuned in the experiments to test their influence on the EDD and ECC. The system architecture also influences the SNR due to:

- noise figure [dB],
- system losses [dB],
- system temperature.

Other system architecture parameters that we do not model since they are too detailed for our purpose:

- receiver architecture (Intermediate Frequency (IF),
- Analog to Digital Converter (ADC))
- Radio Frequency (RF) impairments (*e.g.* phase noise)

Furthermore, the SNR is influenced by the environment or scenario parameters in terms of:

- propagation losses [dB]:
  - path (modeled),
  - distance (modeled),
  - rain,
  - atmospheric conditions,
- fading losses [dB]:
  - multipath/ scintillation (modeled),
  - earth curvature,
  - terrain (modeled),
  - objects.

The parameters that are modeled are appended with (modeled). The propagation path loss on a terrestrial communication path is mainly due to

- multipath effects, which cause large and small scale fading (modeled),
- atmospheric losses.

Large scale fading causes the received power to decrease with distance. Small scale fading causes variations of the signal strength over small distances (wavelengths) due to interference of multipath components. The influence of the environment needs to be estimated. In our model the bit-rate error calculation takes into account the roughness of the sea. We do not take into account rain and atmospheric conditions, as by incorporating the roughness of the sea give us enough possibilities to show the influence of a varying environment on the ocean. Earth curvature is not modeled since the experiments involve entities not further than 40 km apart. Modeling the influence of objects is interesting since it can cause problems in communication, but falls out of the scope of this research.

Throughput and link-stability are also influenced by the probability of packet collision. We follow the model in van Foeken and Kwakkernaat [2011] and do not take packet collisions into account; however, this effect can be incorporated by assuming a constant and independent collision probability. This means that a fixed point collision probability can be determined (Bianchi [2000], Vu and Sakurai [2006]) and as a result the probability of the number of required transmissions can be determined. Packet-delivery error probability is then a combination of packet error (due to channel characteristics) and packet collision (due to collisions with packets send by other sources) probability. Although the collision probability will affect the EDD, its effect on the shape of the distribution. is limited. So we disregard this in our model.

The performance in terms of *latency* can be seen as a second order effect and is influenced mainly by:

- data-rate (bandwidth),
- time-frame duration,
- frame or packet length [s],
- block(packet)-code size [bits],
- guard interval (due to propagation environment, path delay) [s],
- path delay (propagation distance) [s],
- bit-error-rate (BER),
- re-transmissions behavior,

These are all included in our model. Collision probability also influences the latency but as stated, is not included in this model.

In this chapter we use WiMAX as an example communication technique. It is a fully TCP/IP-based terrestrial point-to-point or point-to-multipoint communication system that can serve multiple subscribers over distances of up to 50 km when using stationary, line-of-sight connections. It uses channel bandwidths of 25 MHz to provide data-rates up to 134 Mbps. In the future WiMAX will offer data-rates up to 1 Gbps using enhanced MIMO techniques. It can use a variety of modulation techniques, such as Orthogonal Frequency-Division Multiplexing (OFDM), BPSK, and coding techniques, like convolutional coding, RS coding. For military purposes the NATO has provided NATO band IV (4.4 5 GHz) for WiMAX operation. Military systems based on WiMAX are available and they are capable of sending data up to 37.7 Mbps at 40 km range using 5 Watts of transmit power with up to 64 subscribers (Tel [2008]).

3.4 Communication Model

3.4.1 Expected Delay Distribution

It is assumed that the communication system has knowledge of the run-time expected delay and error probability of transmitting information. In other words, the communication system is able to determine an *Expected Delay Distribution* (EDD), which is a distribution of the probability of successful transmission of a message over time to a set of spatially distributed agents. This distribution is used to determine both the *expected cost of communication* (ECC) as well as the *value* of transmitting the information.

The EDD is calculated based on the *probability* of arrival and the *delay* after successful arrival. Subsequently, probability and delay are combined in a probability delay function, EDD.

To model the probability of delay of transmission by one agent to another agent a, one has to determine the time it takes to transmit the amount of bits that describe the message, $\mathcal{Z}$, and to correctly receive it. Therefore, the SNR at the receiver has to be determined first. The normalized SNR in van Foeken and Kwakkernaat [2011] is given in terms of bit-energy-to-noise-energy ratio (E_b/N_0) as

$$E_b/N_0\left(a\right) = \frac{PGG_a\lambda^2}{\left(4\pi d_a\right)^2 8kT_{\text{sys}}L_{\text{sys}}R} \tag{3.1}$$

where $\lambda = c/f$ with f being the transmission frequency and c being the speed of light. Further parameters are P representing the effective isotropic radiated power of the transmit antenna, G the transmit antenna gain, G_a the receive antenna gain, L_{sys} the system losses, R the data rate, T_{sys} the system temperature, d_a the communication distance to agent a, and k the Boltzmann constant. Here, an average receiver with circuit noise of twice the thermal noise is assumed.

Large scale fading causes the received power to decrease with distance. The loss is calculated according to a specified model. The free-space path loss model [Couch, 1993, eq (4.26)] is defined as $\frac{\lambda^2}{(4\pi d_a)^2}$ and incorporated in the equation. The decay can increase due to interfering ground waves, shadowing and scattering effects. A typical path loss of factor 8 when communicating on sea is reflected in the formula by placing 8 in the denominator.

Data-rate R is related to the bandwidth B. Spectral efficiency or bandwidth efficiency η refers to the data rate that can be transmitted over a given bandwidth, in units [(bit/s)/Hz]. We use the spectral efficiency and bandwidth to calculate the data rate:

$$R = B\eta. \tag{3.2}$$

The value of the parameters G, G_a, L_{sys} and T_{sys} are fixed and dependent on the system, d_a is dependent of the situation, and P and B are parameters that can be tuned by a resource management technique.

By combining E_b/N_0 ratio and small scale fading, the probability of bit-error, q_e, can be determined. Small scale fading influences the bit-error probability because it causes variations of the signal strength over small distances (wavelengths) due to interference of multipath components. Rayleigh fading is fading due to the interference caused to the main signal by the same signal arriving over many different paths, resulting in out-of-phase components incident at the receiver without a dominant component. Rician fading is based on a more dominant component—line-of-sight (LOS)—in combination with other components. The Rician K-factor is defined as the ratio of signal power in the dominant component over the (local-mean) scattered power. Severe fading occurs when $K = 0$ (Rayleigh fading), light fading with a dominant component occurs when $K \to \infty$. Our scenarios in this thesis involve ships observing objects on sea. Particular sea conditions create different Rician K-factors. When waves become higher, the distribution approaches a Rayleigh distribution. See Tab. 3.1 for the relation between sea state and Rician K-factor.

Table 3.1: How Rician K-factor relates to the sea state An [2011]

Wave height	K-factor
$3.5m$ (rough, sea state $= 5$)	$0 < K < 3.5$
$2.1m$ (moderate, sea state $= 4$)	$2.8 < K < 5$
$1.1m$ (slight, sea state $= 3$)	$5 < K$

Rician fading is considered most suitable for a variety of propagation scenarios where Rayleigh fading exists in combination with a strong LOS component. The expected probability of bit-error due to a Rician faded channel using uncoded Binary Phase Shift Keying (BPSK) without diversity can be determined as in Simon and Alouini [2005]:

$$q_e(a) = \left(\frac{1+K}{2+K+E_b/N_0(a)}\right)\exp\left(-\frac{K\left(1+E_b/N_0(a)\right)}{2+K+E_b/N_0(a)}\right) \tag{3.3}$$

The Rician K-factor is to be estimated from the environment and sea-state.

A message has a certain size M. We want to calculate the probability of success when transmitting this message. By communication techniques such as WiMAX a message is often spread over multiple time-frames F. Each time-frame is decomposed in a number of packets Q, each having a size L.

The packet-error-probability, $q_p(a)$, can be determined from the bit-error-probability as

$$q_p(a) = 1 - (1 - q_e(a))^L\,, \tag{3.4}$$

where it is assumed that a single bit error causes a packet error. Similarly, the time-frame-error-probability, $q_F(a)$, can be determined from the packet-error-probability as

$$q_F = 1 - (1 - q_p(a))^Q\,, \tag{3.5}$$

where the number of packets Q is determined as follows:

$$Q = \lfloor t^F/t_p \rfloor. \tag{3.6}$$

t^F is the allocated frame-time and t_p is the packet transmission time:

$$t_p = L/R + t_G. \tag{3.7}$$

t_G represents the guard time. Guard time is used to allow the transmitted data to propagate to the destination node with no interference with other transmissions. That time period is fixed over all the time slots regardless of the distance between the source node and the destination node of the transmission using that slot. It is calculated to be the time needed for propagating a packet over the network diameter, which is the maximum distance between any two nodes in the network.

The probability of a *successful* transmission of the total message, $\mathcal{Z}$, for a single transmission attempt by an agent to agent a is

$$p(a) = \left(1 - q_F(a)\right)^F, \tag{3.8}$$

with F the number of frames needed to send the message:

$$F = \lceil M/(LQ) \rceil. \tag{3.9}$$

For N transmission attempts the probability of successful transmission after the n-th transmission to agent a is

$$p(a, n) = \left(1 - p(a)\right)^{n-1} p(a). \tag{3.10}$$

For multi-unicast transmissions—transmissions of the same message to multiple agents—the probability of successful transmission to all agents $\mathcal{S}$ after N transmission attempts is

$$p\left(\mathcal{S}, N\right) = \left(1 - \prod_{a=1}^{|\mathcal{S}|} p\left(a\right)\right)^{N-1} \prod_{a=1}^{|\mathcal{S}|} p\left(a\right). \tag{3.11}$$

Although all before mentioned parameters are tunable we tune only:

- $P(a)$: a higher power increases the probability of successful transmission.
- B: a higher bandwidth increases the data-rate.
- R: as R is determined by B and η, a higher data-rate increases the E_b/N_0 and eventually decreases the successful message probability $p\left(\mathcal{S}, N\right)$.

Given the probability of successful message transmission, the corresponding latency or delay can be determined. Following the time-line in Fig. 3.2, latency is defined as the time between the moment of initiation, t, of transmitting message $\mathcal{Z}$ comprising Q packets to the sink and the moment of reception, s, of a block acknowledgement confirming the arrival of $\mathcal{Z}$ at the source. When intra-session coding (*i.e* where coding is restricted to packets belonging to the same session or connection) is used Xu et al. [2009], the latency of N transmission attempts to agent a can be subdivided into determinate (transmit) latency, $\Delta t_{\text{det}}\left(a, N\right)$, and random (receive) latency, $N t_{\text{TAT}}$.

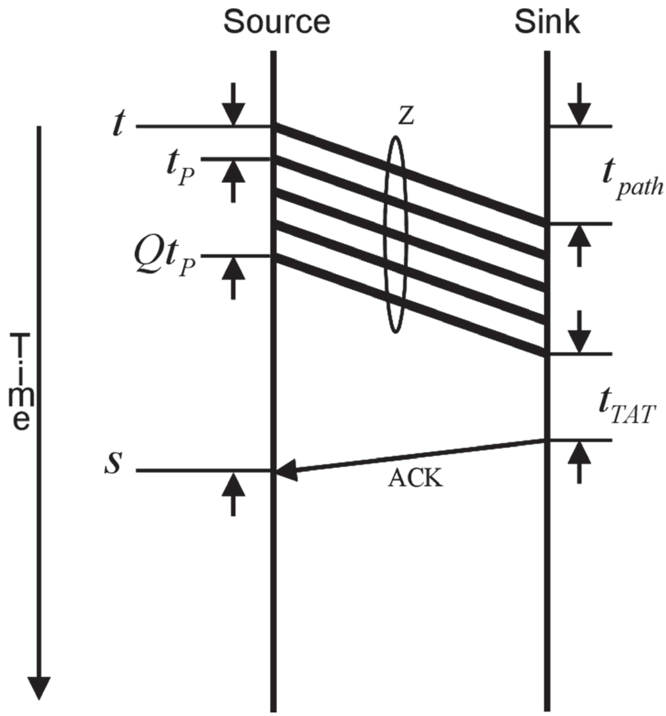

Figure 3.2: A time-line of a single transmission between a source and a sink.

A single transmission frame or multiple transmission frames and a unicast or a multi-unicast influences the formula to calculate the determinate latency. This influence by these parameters on this formula is shown in the following. For a unicast/single transmission frame, latency can be described as

$$\Delta t\left(a, N\right) = \Delta t_{\text{det}}\left(a, N\right) + N t_{\text{TAT}}. \tag{3.12}$$

The first term, Δt_{det}, signifies the determinate or transmit latency,

$$\Delta t_{\text{det}}\left(a, N\right) = Q t_p + 2 t_{\text{path}}\left(a\right) + \left(N-1\right)\left(t_{\text{wait}} + Q t_p + t_{\text{path}}\left(a\right)\right), \tag{3.13}$$

where the number of required transmissions is N, the number of packets Q, the propagation path delay t_p, and the back-off delay t_{wait} due to not receiving an message of acknowledgement (ACK). Here, t_{wait}, is defined as a truncated binary exponential back-off interval.

The second term expresses turn-around-time—TAT(TAT consists of delays in the physical layer for channel-related functions plus processing time for the MAC layer). This is a reformulation of the Round Trip Time defined in van Foeken and Kwakkernaat [2011]. We assume that the TAT follows a Gamma distribution as:

$$p\left(t_{\text{TAT}}\right) = p\left(t - t_{\text{det}}\right) = \Gamma(\alpha, \theta), \tag{3.14}$$

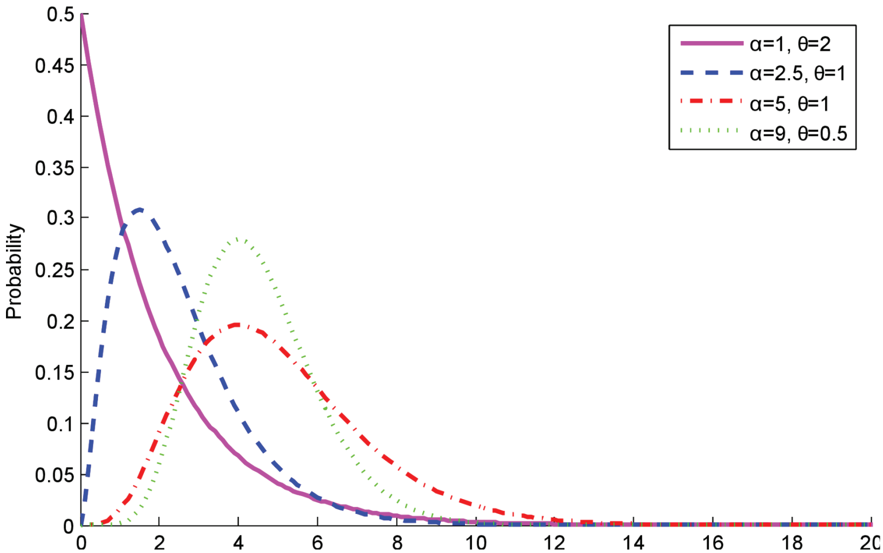

Figure 3.3: Several instantiations of the Gamma distribution.

and is defined in Gunawardena et al. [2003]), see Fig. 3.3 for example distributions. The shape of the distribution is determined by α and the scale by θ. Typical values are $\alpha = 2.5$ and $\theta = 1$ IEE [2006].

For a multi-unicast to $\mathcal{S}$ agents, the single-time-frame latency can be described as:

$$\Delta t\left(\mathcal{S}, N\right) = \Delta t_{\text{det}}\left(\mathcal{S}, N\right) + N t_{\text{TAT}}, \tag{3.15}$$

with

$$\Delta t_{\text{det}}\left(\mathcal{S}, N\right) = Q t_p + 2\max_{\mathcal{S}}\left(t_{\text{path}}\right) + \left(N-1\right)\left(t_{\text{wait}} + Q t_p + \max_{\mathcal{S}}\left(t_{\text{path}}\right)\right), \tag{3.16}$$

where the last term signifies the maximum propagation path delay. When the message is spread over multiple time-frames the determinate latency can be described as

$$\Delta t_{\text{det}}\left(\mathcal{S}, N\right) = 2\max_{\mathcal{S}}\left(t_{\text{path}}\right) + \left(N-1\right) t_{\text{wait}} + t^F_{\text{tot}} F N, \tag{3.17}$$

with the total frame-time duration t^F_{tot} and the number of frames F. The number of frames decreases with a higher data-rate R thus bandwidth B, and therefore decreases the determinate latency. Furthermore, the allocated t^F influences the latency as follows. A frame has a certain total length t^F_{tot}. Resource management allocates a certain fraction t^F of this t^F_{tot} to sending the message and other fractions to send other information. The higher the fraction the more can be put into a single frame, decreasing the number of frames F the message needs to be spread over. Hence, this decreases t_{det}.

When successful message transmission probabilities and latency values are combined, an EDD over time can be constructed for transmission to agents $\mathcal{S}$ over N transmission attempts, $\hat{p}\left(t, \mathcal{Z}, \mathcal{S}\right)$.

During the determinate interval, Δt_{det}, the message will definitely not have arrived at the agents(s). Therefore, during this interval the probability of arrival is zero. After this interval, the probability distribution of a single end-to-end arrival follows the Gamma Cumulative Distribution Function (GCDF), $\mathcal{G}$,

$$\mathcal{G} = P\left(t_{\text{TAT}}\right) = P\left(t - t_{\text{det}}\right) = \int \Gamma(\alpha, \theta), \tag{3.18}$$

(see the blue line in Fig. 3.4).

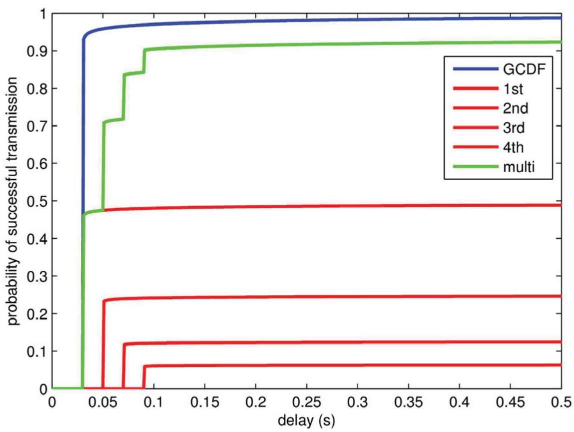

Figure 3.4: In blue, the time-shifted GCDF of the first transmission attempt, $\mathcal{G}\left(t - \Delta t_{\text{det}}\left(a, 1\right)\right)$. The four red lines represent the time-shifted GCDFs multiplied with the probability of success of the first to the fourth transmission attempt. The green line represents the summed ECDDs, calculated by (3.20) with $N = 4$.

For N transmission attempts the GCDF is moved in time by the determinate latency depending on the transmission attempt, multiplied with the probability of success of the transmission attempt and summed for all transmission attempts as

$$\hat{\mathcal{P}}\left(t, \mathcal{Z}, a\right) = \sum_{n=1}^{N} p\left(a, n\right) \mathcal{G}\left(t - \Delta t_{\text{det}}\left(a, n\right)\right), \tag{3.19}$$

where $t_{\text{TAT}} = t - \Delta t_{\text{det}}$. Fig. 3.4 shows an example of these different GCDFs.

The expected cumulative delay distribution (ECDD) of N transmission attempts to agents $\mathcal{S}$, $\hat{\mathcal{P}}\left(t, \mathcal{Z}, \mathcal{S}\right)$, can be calculated as follows: First the product of the separate GCDFs of every unicast to each agent, $a \in \mathcal{S}$, is taken. This is multiplied with the probability of success of the transmission attempt, $p(\mathcal{S}, N)$, to result in the expected cumulative distribution of the nth transmission attempt to agents $\mathcal{A}$. Then, the cumulative distributions of each transmission attempt are summed:

$$\hat{\mathcal{P}}\left(t, \mathcal{Z}, \mathcal{S}\right) = \sum_{n=1}^{N} p\left(\mathcal{S}, n\right) \prod_{a=1}^{|\mathcal{S}|} \mathcal{G}\left(t - \Delta t_{\text{det}}\left(a, n\right)\right). \tag{3.20}$$

This can be illustrated by Fig. 3.4 as well, but then imagine the red lines are the 1st to the 4th multi-cast transmission attempts and the green line is $\hat{\mathcal{P}}(\mathcal{S}, 4)$.

The final step is deriving the EDD by taking the differential equation from (3.20):

$$\hat{p}\left(t, \mathcal{Z}, \mathcal{S}\right) = \frac{\mathrm{d}(\hat{\mathcal{P}}\left(t, \mathcal{Z}, \mathcal{S}\right)}{\mathrm{d}t}. \tag{3.21}$$

The communication service delivers this probability distribution as EDD by request from the evaluation methods.

Another example of an ECDD for a multi-unicast is shown in Fig. 3.5 where the red line is the GCDF over $N = 5$ transmission attempts. The blue vertical lines show the *spiked* ECDD. Each spike represents the probability of the nth successful multi-unicast. Note that the spike delay time is determined when the continuous GCDF of the nth transmission is almost flat (i.e. the probability of a successful nth unicast after this delay is vanishingly small; in this case 0.001). When this moment has been reached the estimation of the ECDD continues with the next transmission attempt. The exponential back-off is simulated by $t_{\text{wait}} = 0.05n^{1.5}$.

In conclusion, the EDD is calculated from the probability of success and delay. A higher amount of power $P(a)$ simply increases the success probability, but the effect of higher bandwidth B is less clear. On the one hand, higher bandwidth decreases the succes probability, on the other hand it decreases the determinate latency. By allocating a higher frame-time fraction t^F, the determinate delay decreases.

3.4.2 Expected Cost of Communication

Evaluation methods determine the reward of sending information based on the value of the data and the expected cost of communicating—ECC. The ECC is defined as the amount of resources used over time, with respect to the total amount of resources available. Here, time is related to the required number of transmission frames, F, times the number of expected transmissions required, N. The expected cost of the nth transmission attempt is then defined as

$$\hat{c}(n, \mathcal{Z}, \mathcal{S}) = Fn \frac{t^F B \sum_{a=1}^{|\mathcal{S}|} P(a)}{t^F_{\text{tot}} P_{\text{tot}} B_{\text{tot}}}, \tag{3.22}$$

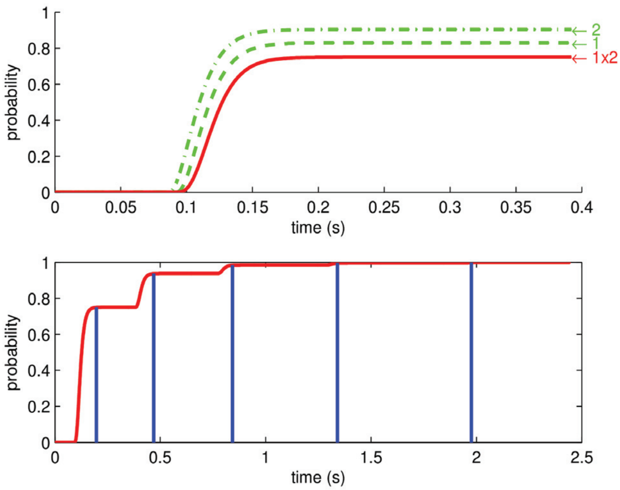

Figure 3.5: The first figure is an example of the probability of successful unicasts of $\mathcal{Z}$ to two entities (in green) and the combined probability of successful multi-unicast to both (in red) versus time-delay. The second figure shows the probability of successful multi-unicast of $\mathcal{Z}$ over multiple transmission attempts. The red line shows the cumulative distribution over multiple transmissions. The blue lines show the spiked cumulative distribution. $Q = 10$, $L_{\text{sys}} = 1000$, $t_{\text{wait}} = 0.05n^{1.5}$, $R_2 = 450kbit/s$, $R_3 = 600kbit/s$, $t_G = 1.0 \times 10^{-3}$, $N = 5$, $t_{\text{path}} = 3.3333 \times 10^{-5}$, $q_e(2) = 1.85 \times 10^{-5}$, $q_e(3) = 1.0 \times 10^{-5}$ and $t_{\text{TAT}} \sim \Gamma(2.5, 0.01)$

and the expected cost of communication (ECC) is the sum of all attempts:

$$\text{ECC} = \hat{\mathcal{C}}(t, \mathcal{Z}, \mathcal{S}) = \sum_{n=1}^{N} \hat{c}(n, \mathcal{Z}, \mathcal{S}), \tag{3.23}$$

It is defined as $\hat{\mathcal{C}} : \mathbb{R}+ \rightarrow \mathbb{R}+$, has a lower bound of zero and increases with the number of frames, F, and number of attempts, n. Here, t^F_{tot}, P_{tot} and B_{tot} represent the total frame length, power and bandwidth. A communication network has a certain total bandwidth B_{tot} available. Assuming that an agent has a certain bandwidth B allocated for transmitting a message to agents $\mathcal{S}$, a larger B claims more of the total bandwidth B_{tot}, hence is more costly. The same applies to the frame length fraction T and the transmission power $P(a)$. Depending on the communication system properties and settings, certain resources, such as bandwidth, can be shared among multiple agents.

Resource management can re-allocate resources such as t^F/t^F_{tot}, B and $P(a)$ to de- or increase the cost. These parameters affect the costs as well as the EDD.

The ECC measures the costs made over time. However, we can compare ECCs more succinctly by deriving an average of the costs:

$$E(\hat{c}) = \sum_{n=1}^{N} p\left(\mathcal{S}, n\right) \hat{c}(n, \mathcal{Z}, \mathcal{S}). \tag{3.24}$$

The function sums the expected costs of each successive transmission attempt, where the expected cost is the probability of successful transmission of attempt n, see (3.11), times the costs for the nh attempt and the attempts before that. $E(\hat{c})$ is a single value expressing the expected use of resources and can be used to compare cost use in different situations. For example, a higher $E(\hat{c})$ means that more costs are expected to be made. The expected costs are influenced by the EDD but also by the costs itself.

3.4.3 Modeling the communication

The communication needs to be fully simulated. This means that next to the EDD and ECC the actual delay and costs of communication must be calculated. The pseudo code for the transmission of a message is given by algorithm 1.

3.5 Simulation and Results

The information-sharing network, which connects all the sensor entities and handles the data exchange, is crucial for the performance of DSS. Highly efficient and interoperable connections that can transmit large amounts of data with low latency are required. We take WiMAX as an example communication technique. In the next section the system parameters of WiMAX are used in the previously presented models as an example. The EDD and ECC are then determined under different conditions.

We considered two examples: firstly an example that showed the impact of changes in the environment on the EDD and ECC, secondly, an example to show the impact of re-allocating parameters such as power and frame-time.

3.5.1 Example I

This example consisted of two parts: first, a part where the sensitivity to changes in the environment and distances between entities influence the EDD and expected costs are shown. Second a part is shown where an agent is under consideration for joining a certain team of agents in sharing identical awareness of an object.

Assume four agents positioned on four ships (entities) team up, $\mathcal{S}_o = \mathcal{A}[a_{o1}, a_{o2}, a_{o3}, a_{o4}]$, to share an *Identical Shared Awareness* of an object o. Agent a_{o1} was connected through link 1 to agent a_{o2} via link 2 to agent a_{o3} and via link 3 to agent a_{o4} We compared the EDDs and ECCs of communicating message $\mathcal{Z}$ by a_{o1} to the other three agents in three consecutive situations:

Situation 1 Agent a_1 was 15 km apart from the other agents and the sea is really rough, $K = 1$

Algorithm 1 Algorithm for the transmission of a message $\mathcal{Z}$

{if transmission of any of the agents $a \in \mathcal{S}$ has a determinate delay, see (3.13), higher than a certain max delay Δt_{max} the message will not be sent}
for $a = 1 \rightarrow \mid \mathcal{S} \mid$ **do**
  if $\Delta t_{\text{det}}(a, 1) > \Delta t_{\text{max}}$ **then**
    print ”*Message can NOT be received within acceptable delay!*”
    return
  end if
end for
{loop over every transmission attempt, n, until the number of expected transmissions required, N}
for $n = 1 \rightarrow N$ **do**
  $t_{\text{TAT}} \leftarrow$ randomdraw $(\Gamma(\alpha, \theta))$ {random draw from gamma distr.}
  {the total delay of the nh attempt is the maximum determinate delay plus the Turn-Around-Time}
  $\Delta t \leftarrow \max_{a \in \mathcal{S}} (t_{\text{det}}(a, n)) + t_{\text{TAT}}$
  {if a random number is lower than the success probability of the n_{th} transmission, $p(\mathcal{S}, n)$, see (3.11) , the transmission succeeds in n transmissions with a delay of Δt}
  if rand$(1) < p(\mathcal{S}, n)$ **then**
    $N_{\text{real}} \leftarrow n$
    $\Delta t_{\text{real}} \leftarrow \Delta t$
    break
  else
    $\Delta t_{\text{real}} \leftarrow$ NaN
  end if
end for
{if the previous forloop ended in a Δt_{real} = NaN the message has not been received by all agents and therefore communication failed}
if Δt_{real} = NaN **then**
  print ”*Message has NOT been received by all agents!*”
else
  print ”*Message has been received by all agents!*”
end if

Situation 2 The sea became somewhat quieter, $K = 4$

Situation 3 Agent a_1 moved closer to the other agents, from 15 to 5 km

Further parameter values are:

1. uncoded BPSK modulation;
2. total power of sender: $P_{\text{tot}} = 15$ W;
3. power of sender to single agent: $P(a) = 5$ W;

4. antenna gains: $G = 3$,$G_{a_2}; G_{a_3}, G_{a_4} = 3$;
5. transmission frequency: $f = 4$ Ghz;
6. system temperature: $T_{\text{sys}} = 300$ K;
7. system losses: $L_{\text{sys}} = 1$;
8. total bandwidth: $B_{\text{tot}} = 25$ MHz;
9. allocated bandwidth: $B = 12.5$ MHz;
10. spectral efficiency: $\eta = 3.7$ (bits/s)/Hz;
11. data rate: $R = B\eta = 46.25$ mbits/s;
12. data message: $M = 1000$kb;
13. packet size: $L = 128$ bytes;
14. guard time: $t_G = 1\mu$s;
15. total frame-time/timeslot fraction: $t^F_{\text{tot}} = 10$ ms;
16. frame-time/timeslot fraction: $t^F = 5$ ms;
17. $t_{\text{wait}} = 0.05n^{1.5}$;

Agent a_1 communicates at a certain moment wants to transmit a message $\mathcal{Z}$. Fig. 3.6 shows the ECDD—top—and the spiked EDD—bottom—in the three situations. The ECDD was clearly worst when the ships were far apart and the sea was rough, given by the blue line; the cumulative probability of success did not reach 0.1 after 1 s. In the bottom figure the attempt probabilities were really low. The green line represents the ECDD with an improved sea state. The probability of success became significantly higher as can be observed from the bottom figure as well. When the ships moved 10 km closer to each other, the ECDD (*i.e.* the red line) had again significantly improved since the probability of a successful first transmission was 0.74 opposed to the 0.35 of the middle situation. All ECDDs slowly converged to 1.

The costs of communication per attempt, $\hat{c}(n, \mathcal{Z}, \mathcal{S})$, were not affected by changing environmental circumstances. However, as the probability of successful transmission of attempt n differed per situation the expected costs of equation (3.24) were also different.

Table 3.2 lists the expected cost, equation (3.24), in all three situations. Unsurprisingly, $E(\hat{c})$ was also worst in the roughest sea state and furthest distance situation. Improving the sea state—$K = 1 \rightarrow 4$—hugely decreased the average costs[2]. Most costs were expected to be made in the first transmission attempt—because the probability, $p(\mathcal{S}, 1)$, of successful transmission was 0.74—when the distance was decreased as well.

[2]decreasing the distance to 5 km instead of improving sea state caused a similar decrease in average costs

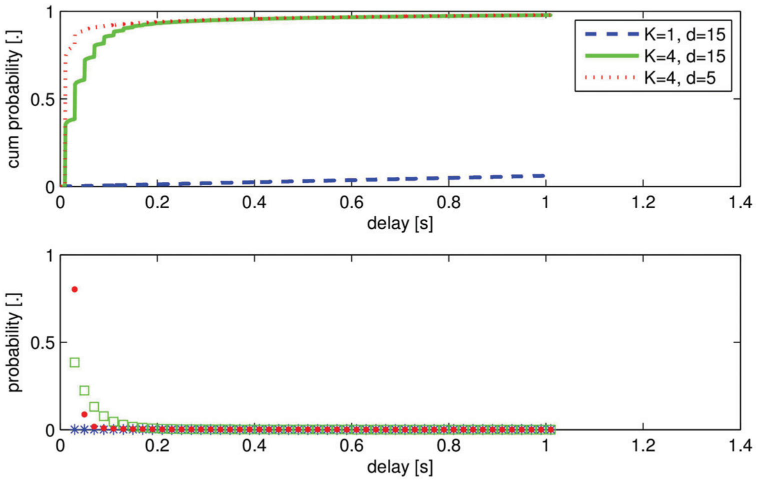

Figure 3.6: ECDD and spiked EDD in three different situations
The top figure shows the ECDDs of communicating to the other team-members.
The bottom figure shows probability of success at each attempt

Table 3.2: The weighted average of costs in different situations

state	weighted cost average
$d = 15$ km, $K = 1$	> 10
$d = 15$ km, $K = 4$	1.66
$d = 5$ km, $K = 4$	1.09

This example shows that the communication model is well able to determine the influence of environmental circumstances on the EDD and the expected costs. By being able to do so, the evaluation methods can adapt to different circumstances. As we will see in chapter 5, RCBE uses the EDD and ECC to evaluate the *expected reward* of sharing detections within the team. They have a significant influence on the *expected reward* and it can for example be possible that a certain detection has a negative reward in situation 1 and a positive reward in—the improved—situation 2.

In the second part of this example we imagined a situation where agent a_1 is under consideration for joining team $\mathcal{S}_o = [a_{o2}, a_{o3}, a_{o4}]$ in sharing an *Identical Shared Awareness* of an object o. Such a consideration is done by ATF described in chapter 6. Agent a_1 determined the EDD and ECC of communicating with the other agents. In the third situation in Fig. 3.6 communication would go quite well. So lets say the agent was included in the team. If later the first situation was at hand, it could be that ATF decided that the agent should be excluded again to improve communication within the remaining team.

3.5.2 Example II

The second example is about the effect of resource allocation on the EDD and the ECC. How can resource management help to improve the communication capabilities? The following parameters are fixed:

1. total power of sender: $P_{\text{tot}} = 15$ W;
2. antenna gains: $G = 3$,$G_{a_2}; G_{a_3}, G_{a_4} = 3$;
3. transmission frequency: $f = 4$ Ghz;
4. system temperature: $T_{\text{sys}} = 300$ K;
5. system losses: $L_{\text{sys}} = 1$;
6. distances to ships: $d_a = 15$ km;
7. Rician K-factor: $K = 4$;
8. total bandwidth: $B_{\text{tot}} = 25$ MHz;
9. allocated bandwidth: $B = 12.5$ MHz;
10. spectral efficiency: $\eta = 3.7$ (bits/s)/Hz;
11. data rate: $R = B\eta = 46.25$ mbits/s;
12. data message: $M = 100$kb;
13. packet size: $L = 128$ bytes;
14. guard time: $t_G = 1\mu$s;
15. total frame-time/timeslot fraction: $t^F_{\text{tot}} = 10$ ms;
16. $t_{\text{wait}} = 0.05n^{1.5}$;

There is a total power supply for communication of $P_{\text{tot}} = 15$ W, and a total frame-time of $t^F_{\text{tot}} = 10$ ms. As we have observed in formulating the EDD and ECC the amount of power allocated for transmission of message $\mathcal{Z}$ has a positive influence on the EDD, but a negative one on costs. Also, the allocated frame-time t^F influences the EDD positively and the costs negatively. This can be shown by example. Again there are three situations:

Situation 1 The power allocated for transmission to a single agent is 1 W so in total 3 W. The frame-time allocated for transmission is 1 ms.

Situation 2 All available power gets allocated to transmission, which means 5 W for each agent.

Situation 3 The whole frame-length becomes available for this message $t^F = 10$ ms.

From Fig 3.7 it can be observed that Situation 1 results in the poorest ECDD, see blue line. This is reflected similarly in the blue stars in the bottom figure that display the succes probabilities of each transmission attempt. By increasing the power the probability of successful transmission of a single attempt improves. The probability of success of a first transmission attempt, see first green square, tops the blue star significantly, and the cumulative success probability is always higher. In allocating more frame-time, the determinate delay t_{det} calculated by eq (3.17), decreases significantly. This can best be observed by the smaller time-intervals between the red dots in the bottom figure. The cumulative success probability is best in this case.

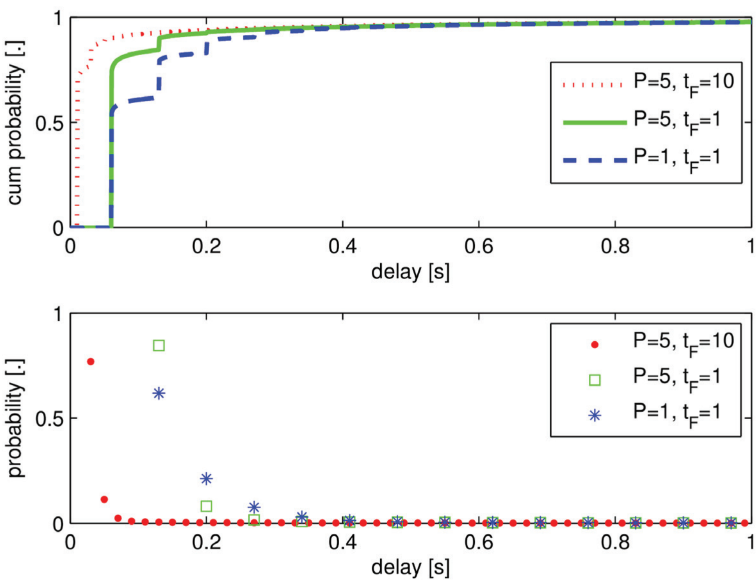

Figure 3.7: ECDD and EDD in three different situations
The top figure shows the ECDDs of communicating to the other team-members.
The bottom figure shows probability of success at each attempt

The costs of a single transmission attempt are displayed in Fig 3.8 for each situation. While the ECDD improves by allocating more resources, the costs do not. Each successive re-allocation increases the costs.

This example is meant to show that the communication model is well able to determine the influence of re-allocating parameters, in this case power and frame-time fraction, on the EDD and the costs. Increasing the allocation of these parameters have a positive effect on the EDD but increase the costs of communication. Both EDD and costs influence, in their turn, the *expected reward*. And depending on other factors, such as the *information-request* this effect is either positive or negative on the *expected reward*. In this regard, the evaluation methods are able

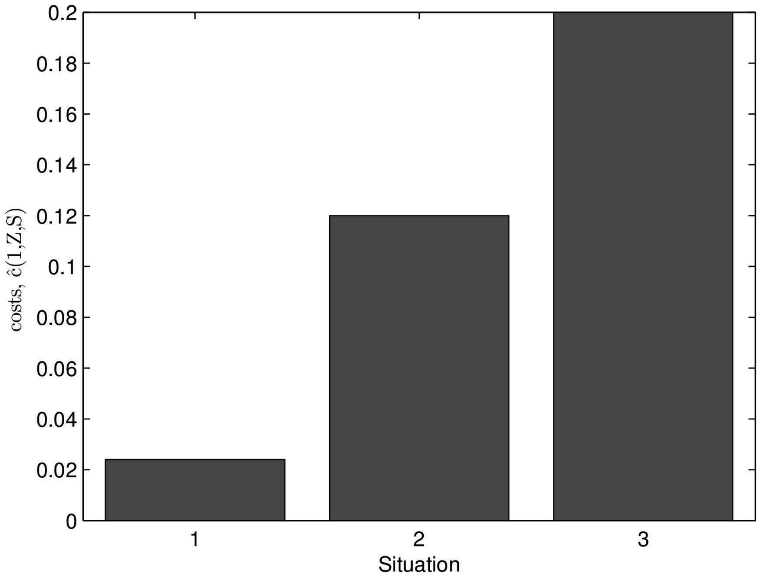

Figure 3.8: Costs of a single transmission attempt.

to find whether increasing power or frame-time, for example, improved the EDD sufficiently to outvalue the increase in costs.

3.6 Conclusion and Future Work

This chapter presented a novel low-complexity communication model of the communication system and the communication channel. The model enabled simulating accurate and up-to-date communication status information—in this case about the expected consumption of communication resources and expected latency. The most dominant performance indicators were identified and their relation to the underlying key parameters was modeled. A formal description of the *expected delay distribution*—EDD—was presented, followed by a formal description of the *expected cost of communication*—ECC. The simulation examples showed that EDDs and ECCs can be estimated for transmitting in different environmental circumstances and transmitting a message to different receiver groups. In addition the effect of reallocating resources on the EDD and ECC can be found. Evaluation methods can use this information to determine whether transmission is rewarding or to determine which entities should join in a team for sharing awareness.

This model does not incorporate the most detailed parameters, but provides the key parameters to enable precise communication information for highly realistic simulations. Such low-complexity models are not present in the literature up till now. This model is novel in that it enables information evaluation methods to be adaptive to changing communication circumstances. You could also experimentally estimate the EDD and fit the parameters of your model. Moreover, the model enables the evaluation of the impact of different communication techniques on the EDD and ECC. And it also enables the evaluation of tuning parameters for resource management?.

The main goal of this chapter was to introduce a generic, low-complex communication model that serves as an enabler for evaluation methods. In addition, this model can be used to quickly compare the performance of different communication techniques in different scenarios. It can even be used to test the run-time adaptation of different communication techniques and decide which one to use. For example, some time-critical information can be sent by a low-latency technique-such as WiMAX, but other less time-critical information by a high-latency technique-such as Link 16.

Chapter 4

Utility and Value of Information

In the introduction we mentioned four approaches that take part in satisfying the *information-requests*. The second one was evaluating the communication capabilities that has been covered in the previous chapter. The third one concerns this chapter:

> approaches that can evaluate the run-time contribution of information to the *identical shared awareness* given the *information-requests*

In this thesis the *contribution* of information is synonymous to the *value* of information for the *Identical Shared Awareness* (ISA). *Value* is defined as *utility* that is gained from using the information for the ISA. In other words, the difference between the utility of the ISA before the information is used and the utility after the information is used. *Utility* is defined as the degree to which the *information-requests* are satisfied. This chapter covers how and why *utility* and *value* are estimated in our system.

Using utility and value in a DSS is useful, since it gives the distributed entities a mechanism to calculate the importance of information for the ISA given the current goals. In other words, goal-directed behavior is enforced. Utility and value of information change with a change of goals. Therefore, it results in an adaptive system. Knowing the utility of the current ISA is useful, but does not tell us anything about how the utility has changed by some gathered sensory information. This change is precisely what we want to obtain, since it enables decision-making about sending or not sending this information to other entities. Value of information is meant to cover this aspect, because it signifies the change in utility that sensory information has brought to the ISA.

In this chapter we formulate an *information-request* as a *utility function*:

> A utility function quantifies the utility as a function of the important information features.

Regarding the features there are two situations possible:

The features contain the information to calculate it This means the value of such a feature can result directly from the estimated state—features like the time of the state or the probability of correct classification.

Additional information is needed to estimate the values of the features Often the feature-values can not be derived from the state directly—features such as the calculation of the error of a track.

The error of a track is not contained in the estimated state itself. To calculate this there needs to be a certain *reference state* to compare with the estimated state. Preferably the reference state is the ground truth. The ground truth is the true state of an object, like the true position or true class or identity. This is only available in test environments like simulation. In a real-world situation and certainly during a run-time process the ground truth is never available. Therefore an estimated reference state is needed to replace the ground truth.

As the ground truth is not available the *utility* and, consequently, the *value* of information cannot be calculated exactly during a run-time process. Therefore the *utility* and *value* can only be estimated and the agents will calculate an *expected utility* and *expected value*.

Although this thesis is all about finding ways to satisfy *information-requests*, how these are exactly formed is out of the scope of this thesis. Nonetheless, to obtain an overall picture, it may be informative to know how *information requests* hypothetically can be constructed in a multi-entity sensor-actuator system, see Fig. 4.1. To begin with, consider a DSS situated in an environment that intends to achieve certain goals. The goal of a group of maritime vessels may for example be to safely move from point A to point B. The group has to watch out for dangers with its sensors and assessment software and crew and possibly manoeuver or act towards objects in order to control the situation.

A DSS that is modeled by our architecture results in objects, which are in this case ships, having a hierarchy of functionalities on the *assessment* (bottom of figure) side and the *management* (top of figure) side, like the JDL model in Steinberg and Bowman [2004]. The goal of this group of ships is to provide *Identical Shared Awareness* (ISA) of the *hostile airborne objects that have been spotted by the ships' radars*. This goal is input for the *situation assessment* level and it reacts by assessing the situation and provides *situation awareness* on the important features of information. For this it needs information about what features of information are important for the *situation management* level to be able to react appropriately to the situation. The *situation management* level determines these features and delivers them back: *accurate tracking information about these hostile fighters*. This is the moment the *situation assessment* level has all the information to set up an *information-request*. From the required *management* features and required *assessment* features it can update the *information-request*: *give accurate information about the location, heading and speed of these objects*.

Given the *information request* the *object assessment* level now tries to improve the accuracy of the tracks of the hostile objects. In this case, all ships have *object assessment* agents that share ISA which has to be improved according to the *information request*.

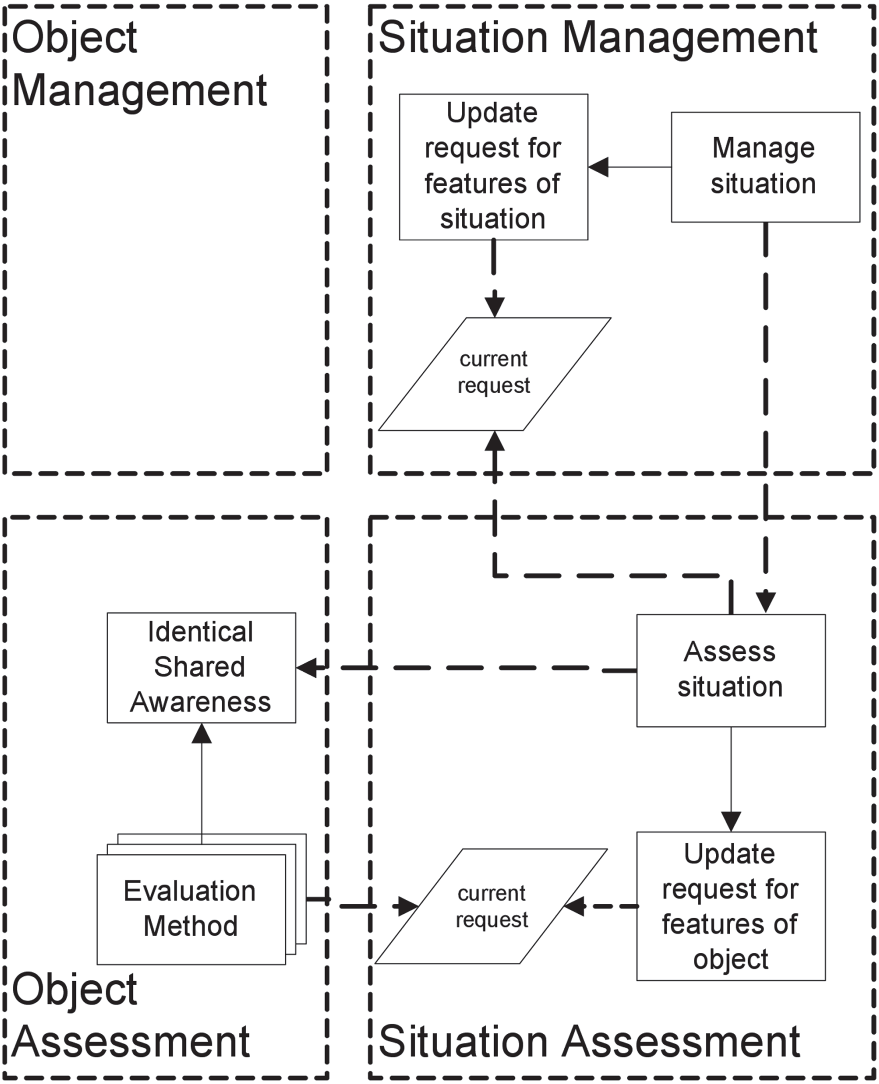

Figure 4.1: Model of a sensor-actuator system
, showing how information-requests for the object-assessment agents can be made.

The chapter starts with section 4.1 that formally defines *information-requests*. Section 4.2 covers the ins and outs of the *reference state*. This is followed by an argumentation of the limitations of information theoretic entropy-based methods and the motivation for and formalization of the novel *integrated utility function* in section 4.3. The last section of this chapter discusses the *value function*. The conclusion finishes this chapter.

4.1 Information-Requests

When a team observing object o, $\mathcal{S}_o$ receives a global *information-request*, the team has to coordinate its awareness according to this *information-request*: sharing locally gathered information to maintain an ISA that satisfies the *information-requests*. Each agent is cooperating in a team with other agents, $\mathcal{S}_o$, to observe and act upon a singular perceived object o—$o \in \mathcal{O}$. This cooperation consists of maintaining an *identical shared awareness* (ISA), $\hat{x}_o$, of the object. An information-request should be seen as a guide for the information provider to determine whether the information it collects is relevant for using for the ISA and so includes the relevant information to determine this.

The DSS uses *information-request*s to improve the construction of *identical shared awareness*. The nature of a certain *information-request* depends on the *domain*, the *information abstraction level*, the *goals* and the current *situation*. The domain here is *Distributed Sensor System*s (DSS) that perform evaluation on the 'object' level of information abstraction. In this thesis, this domain and abstraction level remain fixed but the goals and the situation may change during a mission. The goals of the system are known to request a certain effect on the perceived objects. Depending on the current requested effect, certain features and certain objects will be more important to observe than others, and this will be reflected in the current *information-request*.

Examples of such features are *accuracy* in tracking (Gulrez and Kavakli [2007]), the probability of classification—for example in recognizing a certain person in a large crowd, a *Region Of Interest*—ROI—or *timeliness* (Eswaran et al. [2011]) of information. *Information-requests* may be multi-dimensional in that they signify the utility of multiple features, such as recognizing the person with a probability of 90% (first feature) within a time window of 10 seconds (second feature). Also, an agent can receive requests from multiple agents, for example one agent is interested in information coming from a different ROI than another. Moreover, a team of level $l+1$ agents can express a global *information-request* to a team of level l agents. These level l agents have to coordinate their information flow to fulfill this *information-request*.

The needed state-estimation processing is also reflected in the important features of the information-request. For example, if the information-request features are accuracy of location, speed or acceleration of an object, a tracking algorithm is needed. Or, if it features ambiguity, a specific tracking algorithm that features track probabilities is required. If the information-request requires a certain likelihood of classification of an object, we need a recognition algorithm.

4.1.1 Utility Function

It is important to define what characteristics an *information-request* should minimally have. First of all, an *information-request* posed by a higher information abstraction level needs to have a clear relation with the type of information that the lower information abstraction level produces. The request should quantitatively indicate what and/or when information gathered by the lower abstraction level is relevant. We believe that the *information-request* should be a *utility function*:

> The utility function quantifies the utility as a function of the important features of information.

In this thesis all experiments are focussed on improving the accuracy of the position of some object(s) of interest. Accuracy is defined as:

> The deviation in meters of the estimated location of the object from the ground truth.

The *information-request* therefore represents the *utility* of accuracy of the ISA when it is synchronized. Concerning the accuracy feature of the utility function, there was no real reason that the experiments focussed on the accuracy of position alone, and one can, if that is desired, switch to or include improving other kinematic information, such as velocity or acceleration. A utility function for location accuracy is shown in the left pane of Fig. 4.2, where estimates with an accuracy better than $5m$ are useful and worse are useless. The maximum of the *utility function* represents the most desirable state. Such a maximum or desired state of the *utility function* is used as a reference state in this thesis.

As the moment of expected synchronization is, due to limited communication, delayed, the utility of the estimate of the position at that delayed point in time needs to be calculated. It is evident that without additional information the estimate will degrade and, consequently, the utility as well. In short, the delay of synchronization has a significant influence on the accuracy and therefore on the utility of the estimate. One could argue that because of its influence, the delay should be another parameter in the utility function. However, with tracking algorithms it is possible to forecast—predict—the location of an object in time given the previous trajectory. Consequently, as it is possible to calculate the accuracy at any point in time it is sufficient to have the accuracy as the only parameter. This has as advantage that fewer parameters make the utility function more elegant and less complex for the higher level to compose.

Formally: level 2 agents give *information requests*. An information-request takes the form of a utility function

$$U_{\mathcal{S}}(\Gamma) \in [0, 1]. \tag{4.1}$$

It is defined as $U : \mathbb{R}+ \rightarrow \mathbb{R}+$, and assumed to behave monotonically—either decreasing or increasing. It expresses the utility of any number of currently important evaluation features of information, where the features are expressed, and this is an assumption, as a set of uncorrelated evaluation parameters $\Gamma = [\gamma_1, \gamma_2, \ldots]$. Examples of features that may be important are *region*, *accuracy* or *delay*.

Ultimately, the utility functions are derived from the desired actions in the environment. As discussed in the introduction, a desired action, such as safe passage of ships through a hostile area, requires certain accurate shared awareness of hostile objects. It seems logical that action effectiveness increases when entities cooperate in their action instead of them acting individually. On top of the benefits of sharing awareness, such as increased position accuracy, we incorporate the advantages of action cooperation into the calculation of utility.

This is done by assuming that with the same state, $\hat{x}$, the action effectiveness on objects increases *exponentially* with more entities cooperating. This can be modeled by assuming that the sum of local utilities of a state is smaller than the team utility of the same state. Formally:

$$\forall \mathcal{F} | \cup \mathcal{F} = \mathcal{S} \text{ and } \cap \mathcal{F} = \emptyset : \left(\sum_{\mathcal{H} \in \mathcal{F}} \left[U_{\mathcal{H}} \left(\Gamma \right) \right] \leq U_{\mathcal{S}} \left(\Gamma \right) \right) \tag{4.2}$$

Let, $\mathcal{F}$, be a family of sets where the union of all sets $\mathcal{H}$ of that family equals $\mathcal{S}$ and the intersection of all members is an empty set and $\mathcal{S}$ is the set of all members. For all $\mathcal{F}$, the summed utilities of all sets, $\mathcal{H}$, in the family $\mathcal{F}$ of features Γ, is smaller than the single utility of set $\mathcal{S}$. In other words, each team has a utility function that is a function of the individual subteam utilities: $U_{\mathcal{H}} = f\left(U_1, ..., U_{N_{\mathcal{H}}}\right)$. Such a function is dependent of the specific situation or application, but in this thesis f is defined as:

$$f\left(U_1, ..., U_{N_{\mathcal{H}}}\right) = \left|\mathcal{S}\right|^{\alpha} \sum_{\mathcal{H} \in \mathcal{F}} U_{\mathcal{H}}. \tag{4.3}$$

where $\left(U_{\mathcal{S}} \mid (\alpha > 0)\right) > \sum_{a \in \mathcal{S}} U_a$. α is a variable exponent, where the higher its value the more the effectiveness of a larger team will increase. If α is zero there is no increased effect of coordination.

4.2 Reference State

For calculating the utility $U_{\mathcal{S}}\left(\Gamma\right)$ the *Shell* has to know the values of the features Γ. Remember from chapter 2 that the *Shell* harbors the evaluation methods of an agent, in this thesis Request and Constraint Based Evaluation (RCBE) and Adaptive Team Formation (ATF). As mentioned before these values can result directly from the estimated state, $\Gamma\left(\hat{x}\right)$—for example the time of the state or the probability of correct classification. However, sometimes the utility can not be deduced from the estimated state alone—such as the calculation of the error of a track.

In the second situation the values of the features must then be determined by comparing the estimated state with a certain *reference state*, $\hat{x}_r$, so becomes $\Gamma\left(\hat{x}, \hat{x}_r\right)$. This reference state is for example a desired region of interest (ROI). The reference state can be static, like a ROI, but can also be *dynamic*. This is the case when the agents perform tracking and the desired state from the level 2 agents is a certain *accuracy* of position in the tracks, like in the leading example. Consequently, the reference state has to be deduced from the track estimate at that time. Ideally, the reference state is the ground truth. Subsequently, the error, ϵ, of a track would be determined by calculating the difference with the ground truth of the position of the object. This ground truth is available in simulation, but in reality it is not. Therefore the reference state must be estimated: $\hat{x}_r$.

One possible way of estimating the reference state, $\hat{x}_r$, is by gathering *all* available information associated with the state, $\mathcal{Z}_{all}$: $\hat{x}_r \mid \mathcal{Z}_{all} = \phi(\hat{x}, \mathcal{Z}_{all})$. By using all information that could be associated to $\hat{x}$, the best possible estimate is found. In the tracking example this means the reference state may be the track estimate with all the available detections that could have been associated to the track but have not been used to update the shared track. In a recognition process the reference state may be a probability distribution of several classes given all the local information without any uncertainty. A reference state serves as an absolute comparison for estimated states.

4.3 Integrated Utility Function

A technology that deals successfully with *information-requests* in the system is for example *sensor management* (Kreucher et al. [2005a,b], Bolderheij et al. [2005], Gulrez and Kavakli [2007]), which controls the physical allocation and actions of sensors based on information-requests. Two approaches to sensor-management are task-driven sensor management and information-driven sensor management. *Task-driven sensor management* executes a sensing action that minimizes a currently important performance measure. The performance measure reflects the current request for information, usually for kinematic or identity information, and calculates the error between the true state (ground truth or reference state) and the estimated state.

Information-driven sensor management strives to select an action that maximizes the *information gain*, mostly between the posterior and the prior estimate. Information gain measures like the Kullback Leibler divergence and the more general Rényi divergence (or α divergence) measure the difference between two density functions. For example, Gulrez and Kavakli [2007] introduce a two-step sensor management method where the last step selects the sensor which is most able to deliver information for a user request, by selecting that sensor which has the highest expected information gain. In Grocholsky [2002] an information-based utility function for sensor management is described that can value the information.

In Kreucher et al. [2003] it is argued that Rényi divergence has the advantage over Kullback Leibler—KL— divergence that the α parameter can emphasize different parts of the density. In a follow-up article, Kreucher et al. [2005a], it is claimed that information-driven sensor management has an advantage over task-driven sensor management as it can better deal with different goals when Rényi divergence is used, and therefore actually becomes a task-driven method. By using the Rényi divergence, their experiments showed that the system can better deal with multiple competing goals: in this case a goal for improving tracking accuracy, and a goal for improving identification probability. Moreover, this methodology would be significantly less complex and computationally expensive than task-driven functions. In Aoki et al. [2011b], this methodology was evaluated, and resulted in the conclusion that it has not the same flexibility, as first claimed, towards varying goals as task-driven sensor management.

Next to methods that try to satisfy goals through information gain, one can also *weigh* the information gain. A method doing so is described in Velagapudi et al. [2007], where several policies are presented that aim to maintain a *shared awareness* of the state of the environment in a communication constrained multi-entity team, by locally balancing information value against communication costs. The policies are based on the assumption that entities have no knowledge of each others state and goals. Essentially, a *gossiping technique* (Dimakis et al. [2006]) is used, where sensor readings are, once judged worth sending, transmitted to another *random* entity. Each entity can have different goals, which are defined by a weighted information difference function, in their case the KL divergence. The value of sensor readings differs per entity and depends on its local goals. Value is calculated by the reduction in cost that a sensor reading brings to the estimated state. The cost is a goal-dependant function that weighs the KL divergence between the estimated state and the actual state. Likewise, we used a weighted function (utility function) of the divergence, KL divergence in van Foeken et al. [2009] and Rényi divergence in van Foeken and Kester [2009], for determining the value of sensor information in a tracking application.

In conclusion, the common goal of previously discussed publications was to provide goal directed management of state estimation. The common method was to find the action that maximized the information gain, that measures the difference between two densities. Rényi divergence is able to emphasize different parts of a the density and is therefore more flexible to different goals. Still, as Aoki et al. [2011b] also claim, Rényi divergence as well as KL divergence are not really able to differentiate between different goals. The inherent implicitness of information gain can be somewhat reduced by weighing it according to the current importance. In this thesis we introduce the novel *integrated utility function* that is able to calculate even more explicitly what is relevant, *e.g.* explicit weighting of the uncertainty itself or even different dimensions such as accuracy and timeliness.

Explicit weighting of uncertainty is done by an already existing function, the *Bayes expected utility*. The estimated state is a known probability density function, $\hat{x} = \hat{p}\left(\vec{x}\right)$, of a state variable $\vec{x}$:

$$\mathbf{J}(\vec{x}) \triangleq E\{U\left(u, \vec{x}\right)\} = \int U\left(u, \vec{x}\right) \hat{p}(\vec{x})\mathrm{d}\vec{x} \tag{4.4}$$

It is defined as the expected utility of an action u on state variable $\vec{x}$. An action may be sharing information or a certain sensing action. Bayes expected utility weights the utility gained by the probability of occurrence (an average utility). Every instance of the probability density function (pdf) is evaluated on its utility and weighted with the probability of that value $\hat{p}(\vec{x})$ and results in an average utility. Note that this function is able to calculate the utility of any shape of the pdf.

From this function the *integrated utility function*, $\mathcal{U}$, can be derived, but some changes are needed. First of all, the *integrated utility function* is a function of the important features, Γ, instead of actions. Consequently $E\{U\left(u, \vec{x}\right)\}$ is replaced with $\mathcal{U}\left(\Gamma\right)$. The reference state is a pdf as well, $\hat{x}_r = \hat{p}_r$, of a reference state variable $\vec{x}_r$. To calculate the utility of $\hat{p}$ with respect to $\hat{x}_r$ the utility function can be integrated over all state instances of *both* pdfs. Since it involves two pdfs a double integral needs to be performed. This results in the following 'difference measure', $\mathcal{U}$, between continuous pdf's $\hat{p}(\vec{x})$ and $\hat{p}_r(\vec{x}_r)$:

$$\mathcal{U}\left(\Gamma\left(\hat{p}_r, \hat{p}\right)\right) = \iint U\left(\Gamma\left(\vec{x}_r, \vec{x}\right)\right) \hat{p}\left(\vec{x}\right) \hat{p}_r\left(\vec{x}_r\right) \, \mathrm{d}\vec{x}\mathrm{d}\vec{x}_r. \tag{4.5}$$

This is the *integrated utility function*, and represents a direct and interpretable utility function. All the possible values of the state are directly measured for their utility and probability and lead to a scalar representing the average utility. It is defined as $\mathcal{U} \in [0,1] : \mathbb{R}+ \rightarrow \mathbb{R}+$, and assumed to behave monotonically—either decreasing or increasing.

As an example of calculating the integrated utility, $\mathcal{U}$, we use the utility function in Fig. 4.2: $U\left(\gamma = \epsilon\right)$. This is the utility function as a function of the distance error, $\epsilon = |\hat{x}_r - \hat{x}|$, of tracks—see the left pane of Fig. 4.2: estimates with an accuracy better than 5 m are useful and the utility is zero otherwise. The resulting utilities, $\mathcal{U}$, of different density functions are in right pane. As an example there is a situation where the current track is located at $x = 10$ m. For simplicity it is assumed that the reference state has no uncertainty. Now the information-request can be reformulated: *I need the tracks to have an accuracy that is maximally 5 m*. The three densities represent three situations of the state estimate of the object. The blue density function falls almost completely within 5 m of the reference state, making the utility almost 1. The green density function has exactly the same uncertainty as the blue density but is located some distance from the ground truth; half of the density falls within the 5 m half of it does not. The red density has a far higher uncertainty than the green distribution but still has a higher utility because a larger part of its density falls within the 5 m region.

With respect to information theoretic difference measures we can see the advantages in this example clearly. First, any shape of the density can be measured on its utility. Second, these parts of a density that are utile can be distinguished from those parts that are not; the right half of the green density from the left half; the sides of the red density from the middle. Last, densities with equal uncertainty—the blue and green—can result in different values of $\mathcal{U}$, because it calculates the difference of the estimated state with the reference state.

The difference between $\mathcal{U}$ and information theoretic difference measures, such as KL and Rényi divergence, as explained in this chapter, is that $\mathcal{U}$ takes the important parts of the pdf *directly* into account. With KL this does not happen, and with Rényi only in an indirect and non-intuitive manner. Moreover, Aoki et al. [2011b], show that the following argument for Rényi divergence, the α parameter can be varied to have varying emphases on the tails of the distribution, is tricky and non-intuitive. Another disadvantage of these measures is that *they require the requesting agents to understand the meaning of these measures and adapt their utility function to that*. As they are not easily interpretable, forming information requests becomes difficult. With $\mathcal{U}$ the utility function can have any shape, which is at the same time directly related to the feature(s) that the information-request aims to optimize. Imagine the example in Fig. 4.2. There is no direct way for combining an explicit utility function with an information divergence that reflect the wish for an accuracy of at least 5m.

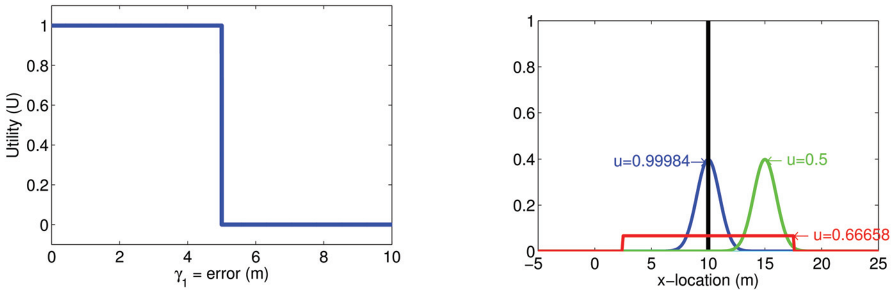

Figure 4.2: (Left) An example of a utility function.
It represents a step function, where an error of tracks of less than 5 meters is useful, and an error of more than 5 meters useless. (Right) This figure shows probability density functions of the reference (the delta distribution at $x = 10m$) and four possible estimates of the position of the object. The utility (using the utility function shown left) of every function is given.

4.3.1 The Value Function

With the *integrated utility function* the *expected utility* of the important features of an ISA can be calculated. But if the impact of detections on these features needs to be known the *gain in utility* that the detections bring to these features needs to be calculated. This gain in utility defines the *expected value*. The ISA providing agents may formulate their *value function* $\hat{\mathcal{V}}(\Gamma, \mathcal{Z})$ as follows:

> A local function for the run-time determination of the expected value $\hat{\mathcal{V}}$ that information $\mathcal{Z}$ has for the relevant features of information Γ, given an information-request from level 2 agents.

Expected value is often expressed as the difference between the reference state and the estimated state, like in task-driven sensor management. Task-driven sensor management, which is one approach to sensor management, executes a sensing action that is expected to result in a measurement that influences the state estimate such that it minimizes a certain currently important performance measure. Let's identify this measure as γ, like our evaluation parameter. The performance measure reflects the current request for information, usually for kinematic or identity information, and finds the error between the reference state, $\hat{x}_r$ and the estimated state $\hat{X}$: $\gamma\left(\hat{x}_r, X\right)$. This performance measure is equivalent to our evaluation parameter, but we use a reference state $\hat{X}_r$ instead of a ground truth.

Aoki et al. [2011a] argue that it may be hard to find a metric that meaningfully measures performance of the estimator, for instance when there are multiple goals with subjective importance. However, as we argue in section 4.1, *information requests* represent the subjective importance of possibly multiple goals. By combining, and this is new, the utility function, U, which represents the information-request, with the evaluation parameters, Γ, we aim to capture subjective importance of goals.

Another approach to sensor management is *information-driven sensor management*, which strives to select the action that will result in a measurement that influences the state estimate such that it maximizes the information gain (or minimize the uncertainty), mostly between the posterior and the prior state estimate. Information gain measures like the Kullback Leibler divergence and the more general Rényi divergence (or α divergence) measure the difference between two state estimates.

Velagapudi et al. [2007] present policies which are able to maintain a *shared awareness* of the state of the environment in a communication constrained multi-component team, by locally balancing expected information value against communication costs. The policies are based on the assumption components do not know anything about one another. Information gain is locally determined and the relevant global information is assumed to emerge from the local interactions.

The method of valuing local information by Velagapudi et al. [2007] inspired our *value function*. Their value function measures the value of detections and several policies, although in different ways, use it to determine transmission:

$$\begin{aligned} \hat{\mathcal{V}}_1\left(s,a\right) = & \\ & C\left(a,\Delta\left(X,P\left(X\left(t\right),t\right)\right),t\right) - \\ & C\left(a,\Delta'\left(X,P'\left(X\left(t\right),t\right)\right),t\right), \end{aligned} \tag{4.6}$$

where $\hat{\mathcal{V}}_1\left(s,a\right)$ is the value of a detection s by agent a. Δ, for example KL or Rényi divergence, is the difference between the actual state X [1] and the state estimate P' over X at time t. They used costs C of information difference, where we use utility. This difference is mostly a matter of taste and does not change the concept. We preferred reasoning with a more *positive* term, *utility*, which determines the usefulness of information. The utility function is actually the inverse of the cost function. Before we go any further with discussing this function let us rewrite the function so that utility is used:

$$\begin{aligned} \hat{\mathcal{V}}_1\left(s,a\right) = & \\ & U\left(a,\Delta'\left(X,P'\left(X\left(t\right),t\right)\right),t\right) - \\ & U\left(a,\Delta\left(X,P\left(X\left(t\right),t\right)\right),t\right), \end{aligned} \tag{4.7}$$

Since $U = -C$, the two terms in Eq. (4.6) are swapped in Eq. (4.7) which makes the two equations equivalent. To continue with Eq. (4.7), U is the utility of the information difference. P' is the local state estimate of agent a. The utility function is a monotonically increasing function; the higher the difference between a state estimate and the actual state the higher the utility. By subtracting the utility of the information difference between predicted (prior) state estimate P and actual state X from the utility of the information difference between the updated state estimate P' and the actual state, the value indicates the improvement of the updated state estimate from the old state estimate. The utility function can be different for every agent and represents its *information request*.

[1] From their article it is unclear what the actual state represents, but we assume that it is the best possible estimate by taking in all available information associated to the state.

This *value function* is, in our opinion, an improvement over the information-driven sensor management way, firstly, because it attempts to find the improvement of the posterior over the prior not by simply taking the information difference between the two, but by comparing both with the actual state, like with task-driven sensor management. By comparing with the actual state the measure becomes *absolute* instead of relative. This argument is exemplified by imagining that the difference between prior and posterior can be the same for infinitely different instantiations of prior and posteriors. By comparing both the prior and the posterior with the actual state, both states can be separately (based on their own probability distribution) measured on their utility. The second improvement is that a cost or utility function is used to represent the subjective importance of the divergence.

Let's show the advantages by an example. There are two value functions, the first represented by equation (4.6) and the second,

$$\hat{\mathcal{V}}_2\left(s, a\right) = U\left(a, \Delta\left(P\left(X\left(t\right), t\right), P'\left(X\left(t\right), t\right)\right)\right), \tag{4.8}$$

which finds the information divergence between the prior state estimate, P, and

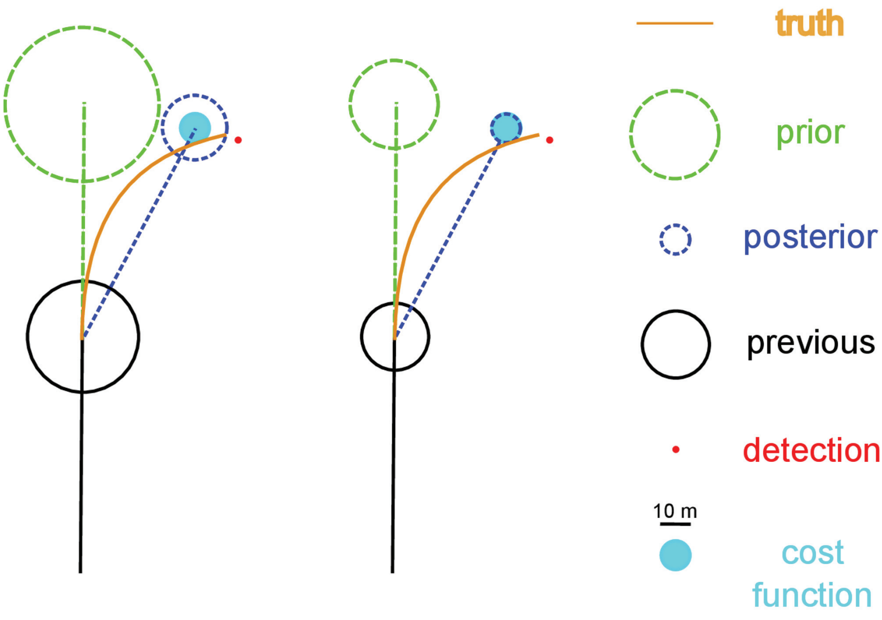

Figure 4.3: Example of two situations.
Both represent a track of an object for which the true trajectory is shown in orange. On your left hand the location accuracy is lower than on your right hand, displayed by the larger black circle. The green dashed line and circle represent the predicted track and predicted accuracy. The blue line and circle represent the track updated with the detection, represented by the red dot. The full blue circle expresses the utility function with a diameter of 10m.

the posterior state estimate, P'. In information-driven sensor management, such a formula is often used and chooses that sensing action which maximizes the information divergence Δ between posterior and prior. [2]

Now imagine there is an agent a observing an object, which is moving to the north. Assume that the 'integrated' utility function $\mathcal{U}$ is:

$$\mathcal{U}\left(a, \Delta\left(P, P'\right)\right) = \iint U\left(\epsilon\left(X, X'\right)\right)\hat{p}\left(X\right)\hat{p}'\left(X'\right)\,\mathrm{d}X\mathrm{d}X'. \tag{4.9}$$

This is the same function as the integrated utility function of Eq. 4.5. ϵ represents the distance between two state instances.

Two possible situations are drawn in Fig. 4.3. Both represent the tracking of an object moving in a curve. When moving in a curve the tracker usually predicts a straight trajectory, hence the large deviation between the updated and predicted state estimate. The black line represents the track before the next detection. All dashed circles represent the uncertainty of the state estimate at time t. Everything within a circle represents 95% of the estimate. The black circle expresses the state estimate before prediction at time $t-1$, the green the prediction state estimate at time t before updating (prior) and the blue dashed circle the state estimate after including the detection (posterior). The blue filled circle expresses the utility function. It has a diameter of 10 m. In the first possible situation, displayed on the left, the uncertainty of the predicted position P_1 is larger than in the second situation, P_2, displayed on the right. In the first situation a detection s_1 is done and in the second a detection s_2. In the first situation the posterior state estimate P_1' is larger than the posterior estimate in the second situation P_2', that is more precise. We assume that the decrease in uncertainty is equal in both situations.

By using the second value function, $\hat{\mathcal{V}}_2$, both situations will result in similar values. The reasoning is as follows. The utility U is only one with an error ϵ smaller than 5m. From the figure it can be seen that both P_1 and P_1' and P_2 and P_2' are far apart. However, there is a larger overlapping area between P_1 and P_1'. Although the areas that do overlap still have low probability the difference causes the integrated utility of P_1' compared with P_1 to be a little smaller than P_2' compared to P_2: $\hat{\mathcal{V}}_2\left(s_1, a\right) < \hat{\mathcal{V}}_2\left(s_2, a\right)$. This seems counterintuitive, since P_2' has a much lower uncertainty and 95% of the estimate falls within the 5m of the utility function.

The problem is that this value function measures the relative difference between prior and posterior. That means that estimates that are far apart in location but have *high* uncertainties will have more overlap, hence lower utility, opposed to estimates that have the same distance but *lower* uncertainties. This way the utility of going from a prior with high uncertainty to a posterior with 0 uncertainty on another position will be higher than when the posterior is less uncertain. Certainly we want the opposite. What we are looking for is a measure that can determine the absolute value/gain in cost of a posterior with respect to the actual state, or in our case the reference state.

With the first value function $\hat{\mathcal{V}}_1$ this is possible. As said, the reference state X is the state with all the information available, which is the posterior state estimate

[2]See equation 3 in Aoki et al. [2011a] for an example of such a formula.

in this case. Taking this into account, eq. 4.6 becomes:

$$\begin{aligned}\hat{\mathcal{V}}_1(s,a) = \\ & U(a, \Delta(X, P(X(t), t)), t) - \\ & U(a, \Delta'(X, P'(X(t), t)), t),\end{aligned} \tag{4.10}$$

where $P(X(t), t)$ represents the predicted state estimate.

95% of P_2' falls within the 5 m region of the reference state. Let's say 60% of P_1' falls within this region. The resulting utilities are then: $U(a, \Delta(X, P_1')) = 0.60$, and $U(a, \Delta(X, P_2')) = 0.95$. Predicted state estimate P_1 has higher uncertainty and therefore will have a slightly higher overlap with the reference state than P_2, and will therefore have a higher utility. Let's have $U(a, \Delta(X, P_1)) = 0.03$ and $C(a, \Delta(X, P_2)) = 0.01$. These numbers result in the following values:

$$\begin{aligned}\hat{\mathcal{V}}_1(s_1, a) &= 0.57 \\ \hat{\mathcal{V}}_1(s_2, a) &= 0.94.\end{aligned} \tag{4.11}$$

This means the second posterior has a significantly higher value than the first posterior, which is what we want to obtain. With respect to the reference state the second posterior is significantly more lowering the cost than the first posterior. This example shows the advantage of $\hat{\mathcal{V}}_1$ over $\hat{\mathcal{V}}_2$.

In conclusion, Velagapudi et al. [2007] captures the difference between the prior and the posterior estimate, between these densities and the actual state/reference state and between subjective costs in one single value function. We have based our approach upon their method.

We take their value function and make the following alterations to finalize our value function:

1. making it a function of the relevant features related to the ISA instead of the agents local state estimate,

2. replacing their utility/cost function with the integrated utility function.

Our goal is to construct ISA, where Velagapudi et al. [2007] do not require this. In addition, we explicitly include the features of information in the formula where they do not. That is, their information difference function Δ suggests that no other features than this, such as timeliness, can be important features of information. This value function intends to be generic for other features of information as well. The value function in Velagapudi et al. [2007] was transformed into (although in slightly different notation):

$$\hat{\mathcal{V}}(\Gamma, \mathcal{Z}) = \mathcal{U}\left(\Gamma\left(\hat{X}_r, \hat{X} | \mathcal{Z}\right)\right) - \mathcal{U}\left(\Gamma\left(\hat{X}_r, \hat{X}\right)\right). \tag{4.12}$$

It calculates the impact of detections on the Γ of information related to the ISA $\hat{X}$ instead of the local state estimate $P^{i'}$. The actual state X by Velagapudi et al. [2007], is replaced with the reference state, $\hat{X}_r$, and is found by gathering all available information associated with the state, $\mathcal{Z}_{all}$: $\hat{X}_r \mid \mathcal{Z}_{all} = \phi(\hat{X}, \mathcal{Z}_{all})$. $\hat{X} \mid \mathcal{Z}$ is the new estimate after including the local information. The *expected value* $\hat{\mathcal{V}}(\hat{X}, \mathcal{Z})$ of some information $\mathcal{Z}$ is expressed as the gain in utility going from ISA, $\hat{X}$, (prior to accounting for $\mathcal{Z}$) to $\hat{X} \mid \mathcal{Z}$. Compared to equation (4.6), this function does not have a parameter pointing to the agent, shares the detection argument $s \equiv \mathcal{Z}$ but has the features Γ as argument instead of the agent.

Finally we inserted the *integrated utility function* as a utility function. $\mathcal{U}(\Gamma)$ replaces $U(\Delta)$, where Γ replaces Δ to indicate the evaluation parameters instead of information difference. The indirectness of the information divergence measure is overcome by this direct and interpretable utility function. All the possible values of the state are directly measured for their utility and probability and the utility function can have any shape.

To return to the example in Fig. 4.2, if the green density is the prior state estimate, $\hat{X}$, and the blue density the posterior state estimate, $\hat{X} \mid \mathcal{Z}$, $\mathcal{U}\left(\Gamma\left(\hat{X}_r, \hat{X}|\mathcal{Z}\right)\right) = 0.99984$ and $\mathcal{U}\left(\Gamma\left(\hat{X}_r, \hat{X}\right)\right) = 0.5$. These utilities would result in a value of 0.49984.

4.4 Conclusion

This chapter introduced the *integrated utility function* that is the method for calculating the utility of information. It was shown how *information-requests* are formed in the system and how they are formalized as *utility functions*. Argumentation was given for the need of a *reference state* to calculate the utility of certain features, like the utility of the error of a track. In case of error the *reference state* is based on all the information that can be associated to the track. Subsequently, several information-theoretic methods for dealing with *information-requests* are discussed but it was concluded they are too indirect and implicit in their measuring of information-utility. This lead to the development of *integrated utility function* that can directly and precisely measure the utility of the important features of information. New is that by combining the utility function, $\mathcal{U}$, which represents the information-request, with the evaluation parameters Γ we are able to capture the subjective importance of goals. The last section presented the *value function* that calculates the gain in *expected utility* caused by new information. We based our approach on the method of Velagapudi et al. [2007] but make it a function of the relevant features related to the ISA instead of the agents local state estimate and replace their utility/cost function with the integrated utility function. Our goal is to construct ISA, where their method does not require this. We showed in an example the advantages of our approach. The *integrated utility function* and the *value function* are central in the functioning of the two evaluation methods presented in this thesis: RCBE and ATF.

Chapter 5

Request and Constraint Based Evaluation

5.1 Introduction

As mentioned in the introduction of this thesis, one of the motivations for developing adaptive methods was to be able to deal better with the increasing desire to fuse lower level information due to the higher complexity of coastal missions opposed to high sea missions. Whereas most research focuses on the fusion of states on the level of tracks or higher, in this thesis it is argued that fusion on the lower plot level increases chances of detection, accuracy and track continuity. However, this change comes with the price that more data is available on this lower level. There are also other problems: Due to changing circumstances in the environment or the system, the quality of communication may change. Importance of certain features might change and sensors might deteriorate. Therefore, flexibility in communication and run-time selection of relevant information may increase the adaptivity to these limitations and enhance the quality of shared awareness.

In this chapter, which is partly based on material from [1], such flexibility is offered for a *Distributed Sensor System* (DSS) that constructs and maintains an *Identical Shared Awareness*—ISA—for objects in the visible environment. An ISA is defined as in 2.2, which in short is an estimated state representing the relevant features of the currently relevant objects in their combined visible environment that, firstly, is *identical*, and, secondly, is *synchronized* on all entities.

The first method of this thesis that offers adaptivity is the *Request and Constraint Based Evaluation* (RCBE) method. RCBE optimizes the ISA by adapting the exchange of local information to the possible changing *information requests* and possible changing communication capabilities. The evaluation algorithm presented here decides whether incoming detections are rewarding to communicate to the other entities and to incorporate into the ISA. This decision is based on the *information-requests*, which are utility functions expressing the current need for

[1] (van Foeken et al. [2009])

information, and communication capabilities of that moment. Certain information can be relevant for one information-request but unimportant for another. Similarly, at one instant in time transmission of information may be too costly and at another not, due to better communication capabilities.

RCBE is a local process estimating the *expected reward* of communicating information to a single or multiple receivers. This reward is a balance of the *expected value* and *expected costs of communication* of the information, and the method decides to communicate when the expected value is higher, and not to communicate when it is lower than the *expected costs of communication*.

Transmission of any information goes through the *Communication Service* (CS). By way of the CS, agent a is able to transmit and to receive detections. The communication capabilities of the network determine how well information can be exchanged. If agent a has information about the expected communication capabilities at that time, it can use RCBE to estimate the *expected value* and *expected costs of communication*. Therefore, we need some communication model for estimating the present capabilities. In this chapter the CS is implemented with the wireless communication model presented in chapter 3. Though the quality of this model does not influence the concept of RCBE, a higher or lower quality does increase or decrease the quality of RCBE's evaluation.

Given the current communication capabilities the CS gives RCBE insight in the current *Expected Delay Distribution*—EDD—and the *Expected Cost of Communication*—ECC.

To show the capabilities of RCBE two questions have to be investigated:

- how well can it adapt to the current request?
- how well can it adapt to the current communication situation?

We have attempted to answer both questions through experiments. These consisted of two sessions:

Accuracy The first experiment session examines the use of RCBE in adapting the communication to changing costs of communication when constructing an ISA that is conditioned by an information request/utility function reflecting the desired accuracy of the location of the fighters at that time.

Accuracy and Latency The second session probes into the adaptivity of RCBE to different severities of the communication capabilities affecting, in addition to changing costs of communication, the delay of communication. Timeliness of information is essential and is indicated by the multi-parameter information request, that next to a desired location accuracy, puts conditions on the latency.

Testing with two different information-requests also demonstrates whether RCBE can deal with different requests.

In a present-day DSS, state-of-the-art communication protocols are used. These communication protocols still naively keep on retransmitting messages. In other words, the message is always being regarded as *valuable* enough to retransmit. In this chapter it is argued that the value of detections can degrade with increasing communication delay. Therefore, determining the *expected reward* of data over time (n retransmissions) may be useful. The decision for retransmission will be determined by the expected reward (e.g. below zero no retransmission).

It works as follows: For every (re)transmission the expected reward can be calculated. After a certain delay the expected reward drops too much for transmission to be relevant—for example, information is more valuable in one second than in two seconds because the system cannot react properly with a long delay. Combining this mechanism with the communication protocol results in a resource and information-request adaptive mechanism.

The chapter starts with a formalization of RCBE in section 5.2, which is given by based on the reward function and the *integrated utility function*. In the next section two sets of experiments are described. The first shows that RCBE causes an increased track accuracy with the same amount of communication opposed to the benchmark method, and the second shows that RCBE is adaptive to different severities of the communication network. The results are discussed in section 5.4 and the conclusion and future work are given in 5.5.

5.2 Formal description of RCBE

We use scenario's where a single team $\mathcal{S}$ of multiple distributed object assessment agents, $\mathcal{S} = [a_1, a_2, a_3, ...]$, receive a global *information-request* for maintaining an ISA of all objects in the visible environment $\mathcal{O} = [o_1, o_2, o_3, ...]$. Because there is a single team the object index o is discarded and we assume team $\mathcal{S}$ maintains ISA of all visible objects—Chapter 6 is concerned with multiple object-teams.

Fig. 5.1 shows a simplified version of the overall process of evaluation in case of maintaining ISA. From left to right: the request handler, the RCBE *Shell* and the communication shell—CS. The request handler updates the current *information request* in case of a new *information request* from level 2. RCBE evaluates incoming *local detections* $\mathcal{Z}$ and calculates the *expected reward* $\hat{\mathcal{R}}$ for the ISA, which is the estimate $\hat{X}$.

The reward $\hat{\mathcal{R}}\left(\hat{X}, \mathcal{Z}, \mathcal{S}\right)$ is calculated by using:

1. the request consisting of a set of utility functions $U(\Gamma)$, with Γ the features of information.
2. the ECC $\hat{\mathcal{C}}\left(t, \mathcal{Z}, \mathcal{S}\right)$ and possibly an EDD $\hat{p}\left(t, \mathcal{Z}, \mathcal{S}\right)$.
3. the local data set $\mathcal{Z}$ and all other available data $\mathcal{Z}_{all}$.
4. the ISA $\hat{X}$.

The ISA is the estimated state $\hat{X}$. RCBE determines which local detections are relevant enough for updating and synchronizing the ISA, hence for communicating them to the other agents in team $\mathcal{S}$. Whether or not detections are relevant is determined by the request, the current EDD and ECC.

The RCBE sub-process performs the following steps:

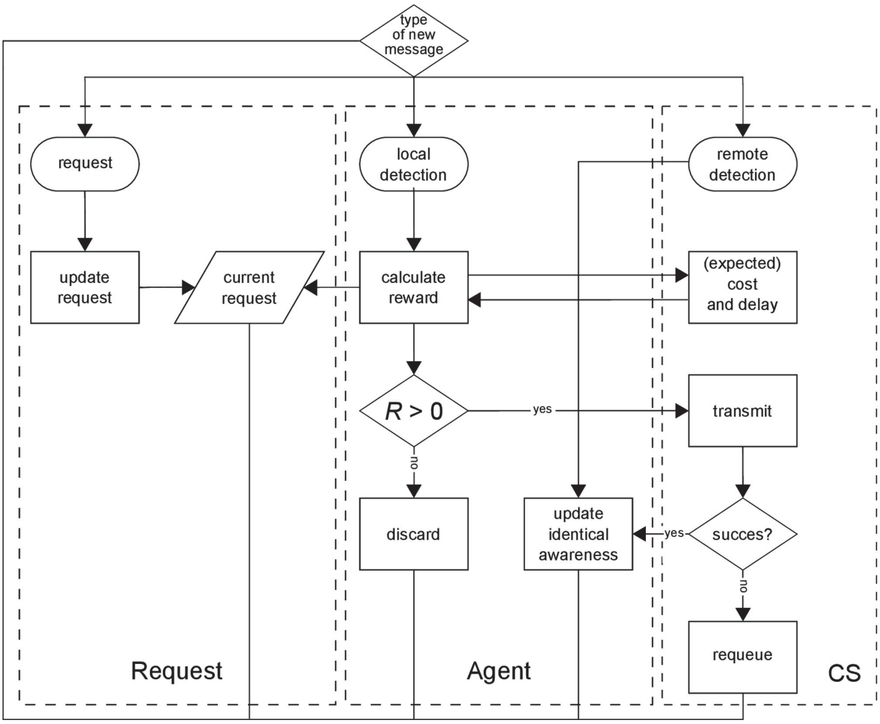

Figure 5.1: Flowchart showing the steps for determining the transmission of detections.
A message enters the agent. The *current request* is updated with the new request. The *reward* of an incoming *local detection* is calculated by the RCBE *Shell*. For this calculation the RCBE uses the current request, the ECC and the EDD. When the reward $\hat{\mathcal{R}}$ is higher than zero, then the CS attempts transmission. If the reward is lower than zero, the ISA is not updated with the detection. If transmission succeeds, the ISA is synchronized by all concerned agents. The ISA is also synchronized by incoming *remote detections*. If transmission fails the data is queued for later evaluation. Hence the loop is closed

- Compute the currently updated mean μ_r of the estimated reference state $\hat{X}_r$.
- The value, $\hat{\mathcal{V}}\left(\Gamma, \mathcal{Z}\right)$ (explained in section 4.3.1).
- The reward $\hat{\mathcal{R}}$ is calculated by subtracting $\hat{\mathcal{C}}$ from $\hat{\mathcal{V}}$
- If the reward is higher than zero, the CS is requested to transmit $\mathcal{Z}$
- When the transmission to the team has succeeded, the ISA is updated with $\mathcal{Z}$ by *every agent in team* $\mathcal{S}$.

When an agent has found relevant information and attempted communication and the agent has received *acknowledgements* from all receiving agents then we assume that the agents can synchronize the ISA and we assume that the ISA is 100% synchronized. Note that in reality communication is never 100% sure and it is difficult to prove that *Identical Shared Awareness* exists. Nonetheless, we believe we can safely make this assumption because the probability of successful transmissions of *acknowledgements* is always close to 100%.

5.2.1 Reward Function

The *expected reward* $\hat{\mathcal{R}}$, that a set of detections $\mathcal{Z}$ brings to the features of information Γ shared between all agents within the team $\mathcal{S}$ is expressed by the following function:

$$\hat{\mathcal{R}}(\Gamma, \mathcal{Z}, \mathcal{S}) = \hat{\mathcal{V}}(\Gamma, \mathcal{Z}) - \hat{\mathcal{C}}\,(t, \mathcal{Z}, \mathcal{S})\,. \tag{5.1}$$

It is expressed as the *expected value*, $\hat{\mathcal{V}}$, that detections $\mathcal{Z}$ bring to the important features of information Γ with substraction of the cost $\hat{\mathcal{C}}$ of communicating $\mathcal{Z}$ by agent a to the other agents of $\mathcal{S}$. A *positive expected reward* results in the transmission of $\mathcal{Z}$ to the concerning agents $\mathcal{S}$ and a *negative expected reward* does not.

For every application it is crucial that $\hat{\mathcal{V}}$ and $\hat{\mathcal{C}}$ are normalized so that they balance fairly. The *value* of information is a manageable variable; it can be lower and higher bounded and in this thesis $\hat{\mathcal{V}} \in [0, 1]$. The *cost* of information, however, is a variable with a growing number, as costs accumulate with an increasing number of transmission attempts. Therefore, cost cannot have a higher bound, which makes proper comparison between value and cost hard, because there is no mathematically sound method to normalize both numbers.

In light of the domain that is the topic of this thesis, however, there is a solution that does make sense. The cost of communicating a message, which is the relevant cost in this thesis was discussed in 3.4.2, is linearly increasing with the product of the number of transmission attempts, n, and number of transmission frames, F:

$$\hat{c}(n, \mathcal{Z}, \mathcal{S}) = Fn\frac{t^F B \sum_{a=1}^{|\mathcal{S}|} P(a)}{t^F_{\text{tot}} P_{\text{tot}} B_{\text{tot}}}, \tag{5.2}$$

A single transmission has a cost that can be normalized. Therefore, a solution can be to normalize the single transmission cost and the value within equal bounds, such as $0 - 1$. Moreover, the cost over multiple transmission attempts can be controlled with a normalization constant κ. If $\kappa = 0.25$ for example, the cost of a single transmission attempt of 1 will drop to 0.25. Four transmission attempts will then be the maximum amount of possible transmission attempts since the upper bound of the value function is 1. Whether this idea holds for many or even every application remains to be seen; that is that the value is bounded and the cost a stepwise increasing number.

The true costs of communication cannot be known at run-time, because the exact communication capabilities in the network changes continuously and is generally not known completely. Therefore, the CS is only able to provide an *expected* cost. The same holds for the value.

The value function, was defined in chapter 4. The agents providing ISA have a *value function* $\hat{\mathcal{V}}(\hat{X}, \mathcal{Z})$ defined as:

> A local function for the run-time determination of the expected value $\hat{\mathcal{V}}$ information $\mathcal{Z}$ has for ISA $\hat{X}$, given an information-request from level 2 agents.

The formula is expressed as follows:

$$\hat{\mathcal{V}}(\Gamma, \mathcal{Z}) = \mathcal{U}\left(\Gamma\left(\hat{X}_r, \hat{X}|\mathcal{Z}\right)\right) - \mathcal{U}\left(\Gamma\left(\hat{X}_r, \hat{X}\right)\right). \tag{5.3}$$

$\hat{X}_r$ represents the (constructed) reference state and $\hat{X} \mid \mathcal{Z}$ the new estimate after including the local data. The *expected value* $\hat{\mathcal{V}}(\hat{X}, \mathcal{Z})$ of some information $\mathcal{Z}$ is expressed as the gain in utility going from ISA, $\hat{X}$, (prior to accounting for $\mathcal{Z}$) to $\hat{X} \mid \mathcal{Z}$.

The utility of information, also defined in chapter 4, is calculated by the Integrated Utility Function (IUF) :

$$\mathcal{U}(\Gamma(\hat{p}_r, \hat{p})) = \iint U(\Gamma(\vec{x}_r, \vec{x}))\, \hat{p}(\vec{x})\, \hat{p}_r(\vec{x}_r)\; \mathrm{d}\vec{x}\mathrm{d}\vec{x}_r. \tag{5.4}$$

Each instance of the pdf $\hat{p}$ is compared with each instance of the reference pdf $\hat{p}_r$ with respect to the features Γ. Each feature value is measured for their utility and integrated to result in a single scalar representing the integrated utility of the $\hat{p}$.

5.3 Experiments

The objective of the experiments was to show that RCBE increases the adaptivity of a DSS. More specifically, to show that, firstly, the DSS can increase the quality of the ISA by using run-time adaptive communication and can cope with requests regarding the accuracy of tracks and the timeliness of information and, secondly, that the DSS can adapt to different communication situations. This increase in quality manifested itself through enabling local agents to perform run-time RCBE. In this thesis we use the communication model and the novel *integrated utility function* to calculate the utility of information. A realistic simulation environment as described in 2.8, the system of van Iersel et al. [2008], is used to run the experiments. The experiments should show how well this communication model and this utility function worked in distributed tracker scenario's.

5.3.1 Scenarios

We performed experiments on two scenarios, shown in Fig. 5.2. Results of experiments on scenario 1 are discussed in section 5.3.2 and of Scenario 2 section 5.3.3.

Both scenario's consisted of a DSS. Scenario 1 comprised two ships j_1, j_2 and scenario 2 comprised three ships j_1, j_2, j_3. The ships used radars to keep objects in their visual range under surveillance and the level 1 agents together maintained

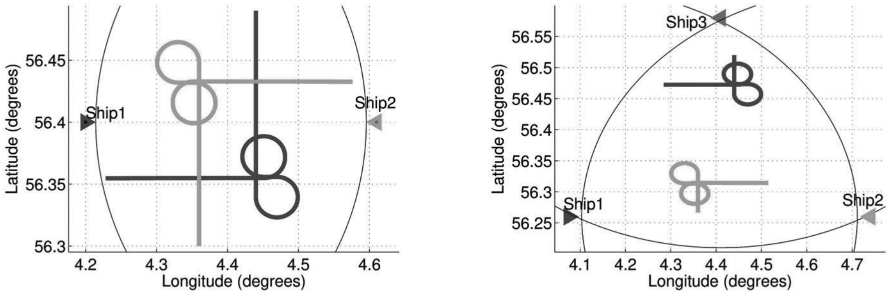

Figure 5.2: (Left) A snapshot of the first scenario in matlab. Two ships with the black circled (overlapping) detection ranges observe two flying objects. The routes of the objects are shown in blue and green. (Right) A snapshot of the second scenario in matlab. Virtually the same as the first scenario, but extended with one extra ship observing the area.

an ISA of the tracks of the objects in the area. Level 0 agents served as a local source of noisy detections. Two unidentified objects, which were in truth hostile fighters, flew into the (overlapping) detection ranges of the two radars and both objects moved in straight lines as well as in loops.

These scenarios represented a DSS that required timely and accurate information about the objects in the environment in a communication constrained network. The ships needed to have an accurate location estimation of the objects, so that the weapon systems could securely eliminate the fighters. However, transmitting detections in high-frequency constrained communication resources, resulted in high communication costs and significant delays. Delay decreased the accuracy of the ISA, hence the effectiveness of the ships on the fighters. So, these scenarios are suited for testing to what extent RCBE can maintain a high quality ISA.

The level 1 agents on each entity were a team $\mathcal{S}$ that received a global information request from the 'situation' information abstraction agents: *Deliver a timely situational picture of the position of the objects in the visible environment. This picture should be equal and synchronized for all entities involved*.

Also part of the information request was a utility function that puts additional conditions on certain features of the state, like track accuracy and communication delay in this article. Other possible features can be region-of-interest or track clarity/ambiguity.

In the core of each agent all locally incoming detections were processed to build up their local tracks. Only the positively rewarded (when the value outweighs the costs) subset of detections were shared with the other team-member and, after communication had been successful, simultaneously processed in a synchronized ISA. Subsequently they delivered it to the higher level agents.

Each ship was equipped with the evaluation process generically shown in Fig. 2.1 and specifically shown in Fig. 5.1. Agents a_1, a_2 and in the second scenario also a_3 maintained ISA of the objects in the area. The $Core$ of the agents performed tracking, and the RCBE *Shell* determined whether level 0 detections $\mathcal{Z}$ were relevant (i.e. $\hat{\mathcal{R}}(\hat{X}, \mathcal{Z}, \mathcal{S})$ is positive) for communicating them and afterwards updating the ISA on all ships.

The tracking algorithm worked as follows: Object detections, $\mathcal{Z}$, described the bearing and range relative to the sensor positions. The tracking algorithm, ϕ, updated the old shared track with new local detections to a new local track: $\hat{X} \mid \mathcal{Z} = \phi(\hat{X}, \mathcal{Z})$.

Every ship harbored a Communication Service (CS) which provides updated versions of the EDD and ECC.

5.3.2 Experiments Session 1

The objective of the first experiment session was to show that a DSS using RCBE increased the quality of the ISA in a communication constrained network over a DSS only using a non-adaptive method like the *Association Method*. We would like to show the influence of bandwidth and took that as the only constraint that influenced the costs of communication. We did not incorporate delays in this scenario and therefore assumed that communication was immediate and always successful. In the experiments with scenario 1, we only used two observing ships because we wanted to show the improvement of ISA already in the minimal setting.

Association Method

The *association method* was a simple method that only shares detections that associate with an existing track. Detections that could not be associated were either false alarms or initiations of new tracks and were put in the local state estimator. If a new confirmed local track (that is when 5 detections are associated to the same new track) was found, the detections of this track were also shared and incorporated in the ISA.

The association method performed a simple run-time evaluation of the costs and did not perform a run-time evaluation of the costs of communication.

RCBE

RCBE was performed as in section 5.2 and shown in Fig. 5.1. Level 2 agents requested accurate information about the location of tracks in the environment. We used as utility function

$$U\left(\gamma_1 = \epsilon\right) = \frac{1}{1 + e^{0.02(\epsilon - 120)}}, \tag{5.5}$$

shown in Fig. 5.3. The evaluation parameter, γ_1, was the error, ϵ. For this illustrative example we used a function that is realistic for surveillance tasks where a rough position estimation of objects is sufficient. The utility function displayed a sigmoid function where the utility is 0.5 when the error of the state is $120m$. The utility slowly increased where an error of 0 had a utility higher than 0.9. With larger errors than $250m$, the utility dropped below 0.1. In a case where the ships need to

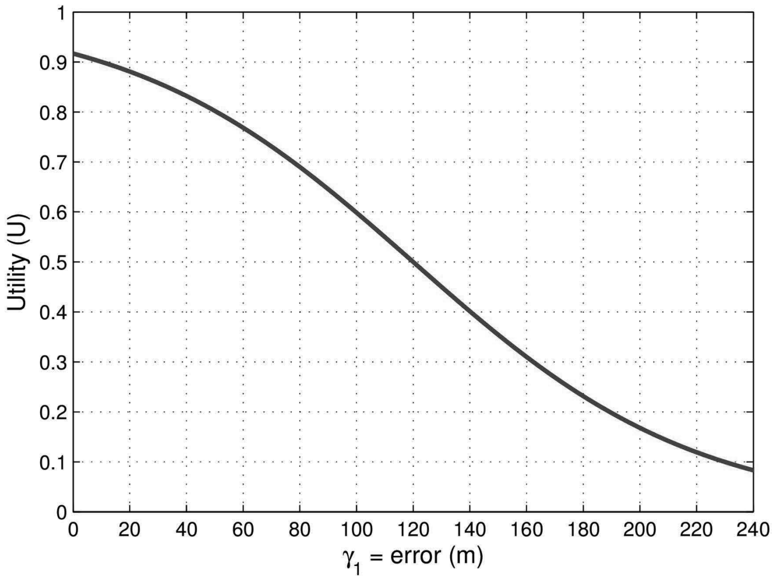

Figure 5.3: The accuracy request of agents a_2 and a_2 take the form of this utility function.

eliminate the objects with their weapon systems, the utility function needs to be aiming for higher accuracy than in a surveillance task.

The RCBE *Shell* only received a cost function $\hat{\mathcal{C}}$ from the CS and together with the request and the agent state $\hat{X}$ it evaluated detections $\mathcal{Z}$. The reward function for agent a_i of communicating $\mathcal{Z}$ within team $\mathcal{S} = a_1, a_2$ was

$$\hat{\mathcal{R}}\left(\hat{X}, \mathcal{Z}, \mathcal{S}\right) = \hat{\mathcal{V}}\left(\Gamma, \mathcal{Z}\right) - \hat{\mathcal{C}}\left(t = 0, \mathcal{Z}, \mathcal{S}\right), \tag{5.6}$$

The value function of $\mathcal{Z}$ for $\hat{X}$ was expressed as the gain in utility going from the prior ISA $\hat{X}$ (without accounting for $\mathcal{Z}$) to agent state $\hat{X} \mid \mathcal{Z}$:

$$\hat{\mathcal{V}}\left(\hat{X}, \mathcal{Z}\right) = \mathcal{U}\left(\epsilon\left(\hat{p}_r, \hat{p} \mid \mathcal{Z}\right)\right) - \mathcal{U}\left(\epsilon\left(\hat{p}_r, \hat{p}\right)\right). \tag{5.7}$$

As described in section 4.3 $\hat{X}$ was actually a probability density function $\hat{p}(\vec{x})$ of the state variable $\vec{x}$ and $\hat{X}_r$ was actually a probability density function $\hat{p}(\vec{x}_r)$ of the state variable $\vec{x}_r$. The reference state replaced the ground truth and to simulate an ideal reference the pdf was a delta dirac function. This resulted in the following reference state: $\hat{X}_r = \delta\left(\vec{x} - \mu_r\right)$ that represented the currently updated mean μ_r of the estimated reference state $\hat{X}_r$ with no error. The integrated utility, $\mathcal{U}(\hat{p})$, was an integration of all the values of the estimated state $\hat{p}(\vec{x})$:

$$\mathcal{U}(\hat{p}_r, \hat{p}) = \int U(\epsilon = x - \mu_r)\hat{p}(x)d\vec{x}. \tag{5.8}$$

On top of simulating an ideal reference, the use of a Dirac function also decreased the computational complexity of the integrated utility function as one integral is removed.

We simplified the cost function of Eq. (5.2), due to the assumption of no delay and a bit-rate error (BER) of 0, to

$$\hat{\mathcal{C}}\left(t=0, \mathcal{Z}, \mathcal{S}\right) = \kappa \frac{B}{B_{\text{tot}}} QL, \tag{5.9}$$

where QL was the number of packets, Q, times the size of a single packet, L, resulting in the data size of detections $\mathcal{Z}$. B was the bandwidth available to the agent, B_{tot} was the total bandwidth in the network and κ was the normalization constant.

Results Session 1

RCBE vs. Association Method To show the increase in performance of RCBE we compared the Mean Squared Error (MSE) of the resulting tracks when using the *association method* versus *RCBE*. To compare the two methods, the amount of communication in both cases needed to be the same.

The Association Method To manipulate the amount of communication with the association method, the CS limited the number of associated detections to be communicated to the required percentage. For example: when in a session with RCBE, 50% of all detections were communicated. In the association method every other associated detection was communicated. We assume that the large majority of detections are associated to tracks, 10 runs were performed where the percentages of communication were varied between 10% and 70%.

RCBE To simulate varying communication situations we varied the bandwidth B in order to vary the ECC, (5.9). When the ECC increased, the value needed to be higher in order for detections to be rewarding, hence to be communicated. We performed another set of 10 runs, and varied the bandwidth to cause the same spectrum of communication percentages as in the experiments with the association method.

Fig. 5.4 shows that for all percentages of communication, RCBE resulted in a lower MSE, hence a better track quality. In addition RCBE performed increasingly better with decreasing percentages of communication, as could be expected. Another straightforward observation is that RCBE communicated less when the costs of communication were higher.

Every new positively rewarded detection caused an update of the ISA. Fig. 5.5 shows the update moments of the ISA in a random run, when using RCBE. The frequency of update moments in corners was higher than on the straight parts (i.e in corners $0.5/s$ versus $0.33/s$ on the straight parts).

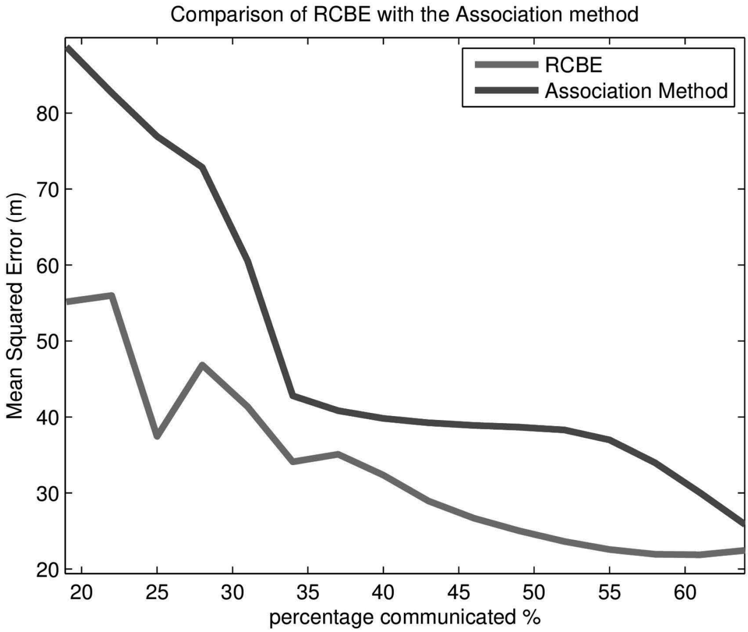

Figure 5.4: Two graphs which show the Mean Squared Error of the tracks versus the percentage of communicated detections for a single run. The blue line shows the MSE when using the association method and the red line the MSE when using RCBE.

5.3.3 Experiments Session 2

The objective of the second experiment session was to show that a DSS using RCBE increased the adaptivity to latency and error in communication. Chapter 3 describes that the reward of detections possibly decrease with longer delays (e.g. the detection in this message is outdated, because for example the ISA can already be updated with measurements that are received by all ships). Therefore, reevaluating detections that failed to be transmitted can be useful. In these experiments we wanted to show that when the global information-request demands timely information, performing reevaluation of detections resulted in communication that is adaptive to different levels of severity in the communication situation.

We experimented on scenario 2. We used three ships because the minimally sufficing setting to enable multi-unicasts is three entities. Each level 1 agent in team S determined the reward of each local detection, where the utility function has two arguments: the current track accuracy and the expected communication delay. The CS was requested to perform a multi-unicast for every positively rewarded detection. The CS kept on attempting to transmit and retransmit the message until RCBE results in a negative reward of the message or the message was successfully

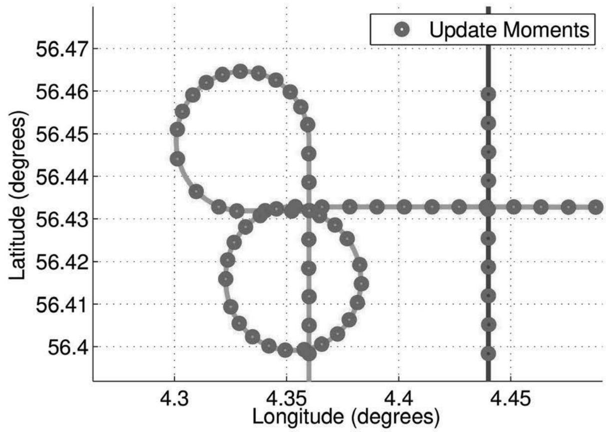

Figure 5.5: A snapshot of the scenario.
The red dots show the update moments of the ISA. When the objects moved in a curve the ISA state is more frequently updated than when the objects move in a straight line

transmitted.

Level 2 agents requested timely and accurate information from team $\mathcal{S} = a_1, a_2, a_3$ about the location of tracks in the environment. Similar to the first experiment session, this utility function is realistic for surveillance tasks where a rough position estimation of objects is sufficient and incorporate both the costs and the delay:

$$U(\gamma_1 = \epsilon, \gamma_2 = \Delta t) = \frac{1}{1 + e^{0.02(\epsilon - 120)}} \frac{1}{1 + e^{5(\Delta t - 1)}}, \tag{5.10}$$

Fig. 5.6 displays the function, where ϵ is the error in meters and d the delay. The first fraction equals the utility function of the first experiment session Eq.(5.5). In addition, the delay is incorporated; the utility dropped while the delay increased, and after two seconds, the most accurate information was useless.

We modeled the communication between ships in line with the communication model described in chapter 3. This section describes the determination of the EDD in detail as well as the cost function. RCBE required the EDD and the Cost function from the CS, the requests from the level 2 agents a_2, and its state $\hat{X}$ to calculate the reward of incoming detections.

The bandwidth for each agent, B, was kept constant and the bit-error rate BER was varied to initiate different degrees of severity of the network. For the experiments we simplified the continuous EDD to a spiked EDD. The lower part of Fig. 5.7 shows an example of a continuous EDD (in red) and a discrete spiked EDD (in blue). The probability density function of the latter discrete GCDF is shown in Fig. 5.8 and was used in the experiments. The spikes were non-equidistant and ev-

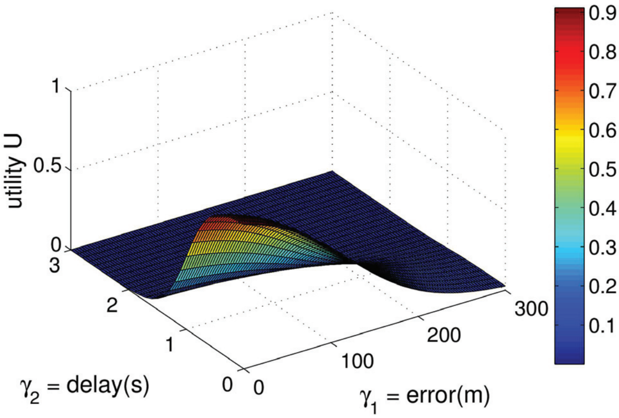

Figure 5.6: An example of a utility function.
The consumer requests information regarding the position of the object. The utility of knowledge about a state estimate depends on the error of the estimate as well as the delay in receiving this estimate. If the knowledge is perfect, i.e. zero error and zero delay, the utility is the highest

ery transmission had an equal success probability. The Expected Delay Distribution (EDD) function was

$$\hat{p}\left(t, \mathcal{Z}, \mathcal{S}\right) = \sum_{n=1}^{N} p\left(\mathcal{S}, n\right) \prod_{a=1}^{|\mathcal{S}|} \mathcal{G}\left(t - \Delta t_{\text{det}}\left(a, n\right)\right). \tag{5.11}$$

for a multi-unicast transmission schedule of N transmission attempts and transmitted to the other agents in team $\mathcal{S}$ of message $\mathcal{Z}$. $\Delta t_{\text{det}}\left(a, n\right)$ is defined as the transmit latency, see section 3.4 for further explanation. $\hat{p}$ is the cumulative distribution of a multi-unicast transmission. The probability of a successful nth multi-unicast transmission is:

$$P\left(\Delta t_n\right) = \hat{p}\left(\Delta t_n, \mathcal{Z}, \mathcal{S}\right) - \hat{p}\left(\Delta t_{n-1}, \mathcal{Z}, \mathcal{S}\right) \tag{5.12}$$

with $\Delta t_n > 0$ and $P \geqslant 0$. $P(\Delta t_n)$ is the probability of success after the $(n-1)$th transmission failed.

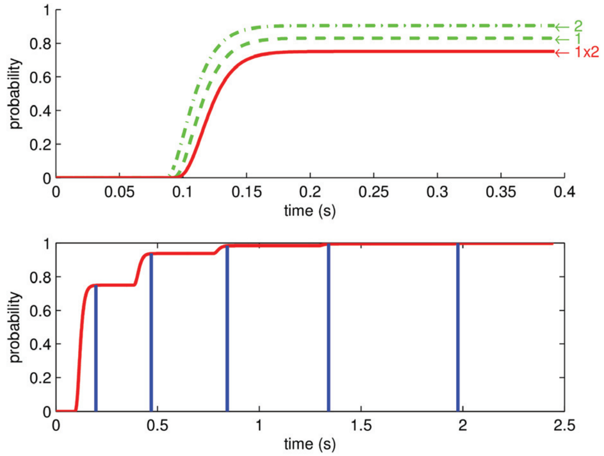

Figure 5.7: The first figure is an example of the probability of successful unicasts of $\mathcal{Z}$ to two entities (in green) and the combined probability of successful multi-unicast to both (in red) versus time-delay. The second figure shows the probability of successful multi-unicast of $\mathcal{Z}$ over multiple transmission attempts. The red line shows the cumulative distribution over multiple transmissions. The blue lines show the spiked cumulative distribution. The first spike shows that there is a 0.74 probability that the message will arrive at the two entities after the first transmission attempt. $Q = 10$, $L = 1000$, $t_{\text{wait}} = 0.05n^{1.5}$, $R_2 = 450kbit/s$, $R_3 = 600kbit/s$, $t_G = 1.0 \times 10^{-3}$, $N = 5$, $t_{\text{path}} = 3.3333 \times 10^{-5}$, $q_e(2) = 1.85 \times 10^{-5}$, $q_e(3) = 1.0 \times 10^{-5}$ and $t_{\text{TAT}} \sim \Gamma(2.5, 0.01)$

Reward function

Using a spiked EDD like in Fig. 5.8, the reward function becomes:

$$\hat{\mathcal{R}}\left(\hat{X}, \mathcal{Z}, \mathcal{S}\right) = \sum_{n=1}^{m} P\left(\Delta t_n, \mathcal{Z}, \mathcal{S}\right)\left(\hat{\mathcal{V}}\left(\Gamma, \mathcal{Z}\right) - \hat{\mathcal{C}}\left(\Delta t_n, \mathcal{Z}, \mathcal{S}\right)\right) - \hat{q}\sum_{n=1}^{m} \hat{\mathcal{C}}\left(\Delta t_n, \mathcal{Z}, \mathcal{S}\right). \tag{5.13}$$

After a certain delay Δt_m the reward drops below zero. Only the m positively rewarded spikes are summed ($m \leqslant n$). Included in the reward is the possible costs made after a failure of transmission, hence after a delay of ($m \leqslant n$) seconds. The probability of this failure, $\hat{q}$, is multiplied with the communication costs made up to Δt_m. These expected costs, $\hat{q}\sum_{n=1}^{m} \hat{\mathcal{C}}\left(\Delta t_n, \mathcal{Z}, \mathcal{S}\right)$. $\hat{q}$ is given by $\hat{q} = 1 - \hat{p}\left(\Delta t_m, \mathcal{Z}, \mathcal{S}\right)$.

The value function is expressed as the gain in utility going from ISA $\hat{X}$ (without

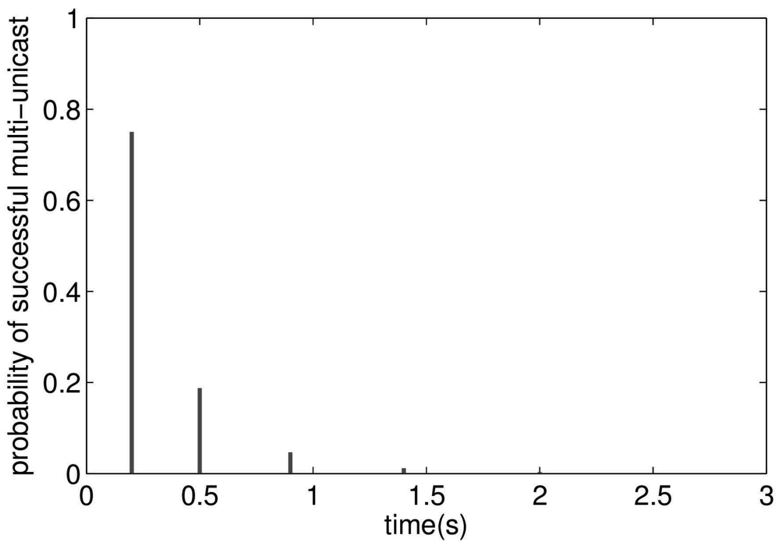

Figure 5.8: One of the spiked delay distributions used for the experiments. Every multi-cast transmission has a probability of 0.75 to succeed. The delay between spikes increases exponentially.

accounting for $\mathcal{Z}$) to ISA $\hat{X} \mid \mathcal{Z}$:

$$\hat{\mathcal{V}}\left(\Gamma, \mathcal{Z}\right) = \mathcal{U}\left(\epsilon\left(\hat{p}_r, \hat{p} \mid \mathcal{Z}, \Delta t\right)\right) - \mathcal{U}\left(\epsilon\left(\hat{p}_r, \hat{p}, \Delta t\right)\right). \quad (5.14)$$

To approach an ideal reference and to decrease the computational complexity, the reference location state is again the currently updated mean μ_r of the estimated reference pdf $\hat{p}_r$ with no error. The reference delay state is the time t, that detection $\mathcal{Z}$ is observed. The actual utility of $\hat{p}$ was an integration of all the values of the estimated state $\hat{X}$:

$$\mathcal{U}(\hat{p}_r, \hat{p}) = \int U(\epsilon = \vec{x} - \mu_r, \Delta t = \Delta t_n - t)\hat{p}(\vec{x})\Delta t\vec{x}. \quad (5.15)$$

The CS provided an ECC for the RCBE shell. This ECC reflected the expected cost for a single spike delay of a spiked delay distribution. It depended on the determinate latency Qt_p of a spike and the fraction of the total bandwidth B_{tot}:

$$\hat{\mathcal{C}}\left(t, \mathcal{Z}, \mathcal{S}\right) = \kappa \frac{B}{B_{\text{tot}}}(Qt_p), \quad (5.16)$$

where Q was the number of packets. Here, the determinate latency Qt_p influenced the cost, because in this experiment session, unlike in experiment session 1, delay was modeled. The determinate latency is costly because during this time the CS was unable to send other information.

Results Session 2

The objective of the first experiment in this session was to show the influence of delay and retransmissions. Fig. 5.9 shows the result of this first experiment. For the first, second and third attempt it shows the accumulation of successfully transmitted detections. The black plot shows the total number of positively rewarded detections before the 1st attempt, hence includes the number of failed transmitted detections. This figure shows that detections were still valuable enough to send after a second and even a third transmission attempt. It also shows that most detections that failed to be send after the second attempt, were irrelevant to be send a third time. Moreover, never was a detection relevant enough for a fourth attempt. When the delay increased this had a double negative effect on the value. Firstly, the *error* between the reference state $\hat{X}_r$ and the estimated states $\hat{X} \mid \mathcal{Z}$ and $\hat{X}$ grew, because the ϵ of the pdfs of the tracks increased. Subsequently, this decreased the $\mathcal{U}$ of both states, and most importantly the difference between these $\mathcal{U}$'s, i.e. the value. Secondly, the delay itself decreased the value. The delay also increased the costs. Apparently the cost always outweighed the value of de detections after the fourth attempt, i.e. $1.4s$. The black plot indicates exactly the phenomenon that several detections, which initially were relevant, decreased so much in value and increased so much in costs that the reward dropped below zero and prevented the detections from being transmitted.

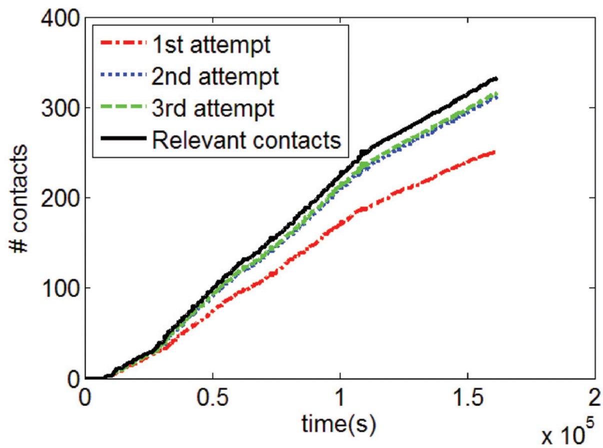

Figure 5.9: Graph showing the cumulative number of detections send after the first (red), second (blue) and third (green) transmission attempt. The black line shows these detections that have been evaluated relevant for the first transmission.

The objective of the second experiment was to show RCBEs' capability of dealing with a varying communication limitations. Firstly, a run was done where communication was artificially forced to be successful after a delay of 0.2 seconds, and resulted in a cumulative number of successful transmissions depicted by the blue line Fig. 5.10. More realistic communication settings where tested in two other runs, represented by the red and the green line. The first setting includes the EDD of Fig.5.8. The second setting has the same delays for the spikes. The bit-rate errors of transmitting to the other two agents changed to $q_e(2) = 3 \times 10^{-5}$ and $q_e(3) = 2 \times 10^{-5}$, causing the probability of successful multi-cast transmission to be 0.6. Rightfully, the red and the green line show that less success probability of communication resulted in less communication. This is because RCBE incorporated this chance of failure, $\hat{q}$, in calculating the expected reward (see Eq 5.13), which lowered the expected reward and therefore decreased communication. This behavior is what we intended, because the costs should increase and the value should decrease due to worse communication capabilities. Another phenomenon was that the lines have a steeper slope in the middle than in the beginning and the end of the run. This means the rate of communication was higher in the middle of the run and was due to the fact that the RCBE *Shell* correctly finds more relevant detections when the objects moved in curves.

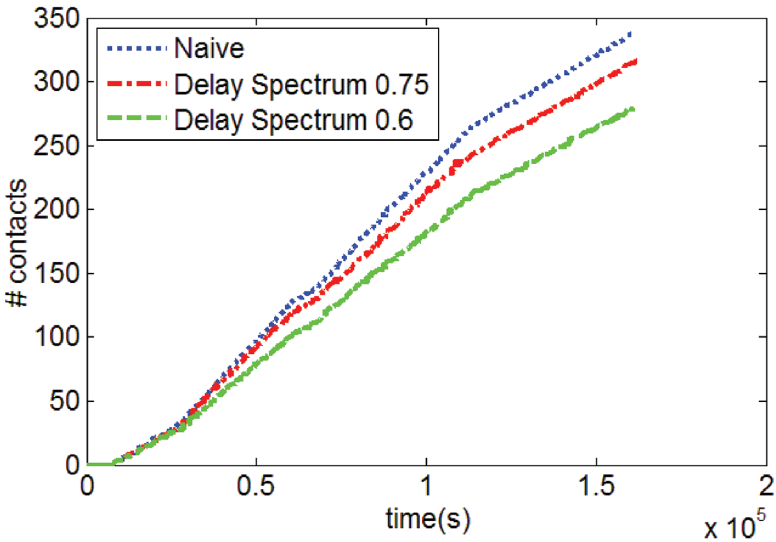

Figure 5.10: Graph showing the cumulative number of detections for an EDD where the probability of a successful multi-cast transmission was 1 (blue), 0.75 (red) and 0.6 (green)

Fig. 5.11 shows the reward of every detection over time in one of the runs with the 0.75 EDD. It shows that the smoothed reward was lower in the beginning and end of the run, exactly where the objects were moving in a straight line, than in the middle period of the run where the objects were making circles.

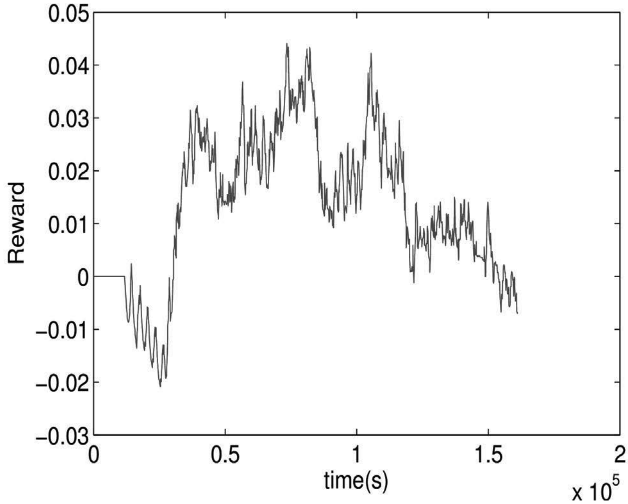

Figure 5.11: Graph showing the smoothed reward of all detections over a single run with an EDD where the probability of successful transmission was 0.75.

5.4 Discussion

We performed two experiment sessions. The first experiment session consisted of a comparison of the *association method* and RCBE. Fig. 5.4 shows that the DSS using RCBE resulted in a better track quality, i.e. ISA, than the DSS when using the association method. The agents, achieved a higher quality of their ISA with the same amount of communication, because RCBE enabled them to distinguish the relevant detections from the irrelevant ones.

The increase in track accuracy was, firstly, due to the *higher* update rate of the ISA state in case the objects were curving and, secondly, due to a *lower* update rate in straight parts. The value of detections was higher in curves due to a *more rapidly increasing* difference between the integrated utility of the updated state estimate and the integrated utility of the non-updated state estimate. This more rapidly increasing difference was mainly because the non-updated ISA deviated faster from the reference state when objects were curving.

In (van Foeken et al. [2009]) we performed similar experiments on exactly the same scenario. The main two differences were that we now used both a novel utility function and a more grounded communication model. The novel integrated utility function is a more direct and intuitive function for estimating the utility of probabilistic state estimates than the utility function we have used in our previous articles. That the new RCBE method resulted in similar outcomes means that the *integrated utility function* is, as well as the old utility function, able to distinguish relevant information from less relevant information, which shows the robustness of our approach. We can conclude that the RCBE method is applicable in a scenario where multiple ships maintain ISA of multiple objects. Moreover, it is able to adapt the communication to the request for accurate location information and to varying bandwidths.

The objective of the second experiment session was to show that RCBE helps agents to adapt the communication simultaneously to latency and error constraints, and to an information-request for a timely and accurate ISA. We have experimented using the communication model described in chapter 3 to realistically calculate both Expected Delay Distributions (EDDs) and an Expected Cost of Communication (ECC). The results show that RCBE enabled agents to adapt their communication to different EDDs; agents communicate less when the expected probability of success was lower. The results also show that it was valuable to reevaluate detections after a failed transmission, because detections often became irrelevant. The decreasing reward can be explained by both a decreasing value of the detections and an increasing cost of communication. Moreover, the agent might have received more detections during this delay interval, which in all likelihood decreased the value of the earlier detection. The results show, in this specific scenario and chosen reward function, that reevaluating detections after failed transmission attempts is valuable. In these experiments RCBE shows that it can deal with varying communication capabilities. Although these experiments have included a limited set of communication settings, it is expected that RCBE will produce resembling results in a wide variety of other communication circumstances.

5.5 Conclusion

The objective of this chapter was to show that RCBE enables distributed entities to maintain Identical Shared Awareness (ISA) and to run-time adapt their interaction to both the current information-requests and the current constraints due to communication. We have assumed that ISA is guaranteed when the transmitting entities receive acknowledgement messages from all the receiving entities. It follows that the DSS can perform coordinated actions based on this ISA.

We have done simulation experiments on two maritime scenarios, where multiple ships are instructed to maintain an ISA of the flying objects in their overlapping detection ranges. The results show that RCBE enables the ships to run-time adapt the communication to two different requests: a request for a certain *accuracy of tracks* and a multi-dimensional request for both a certain *accuracy of tracks* and *timeliness* of information. Otherwise stated, RCBE can cope with varying requests.

The results also show that RCBE simultaneously enables entities to adapt to varying levels of communication capabilities, *i.e.* varying data rates and varying bit-rate errors. The ISA improved significantly when the DSS used RCBE compared to the ISA when the DSS used the benchmark association method. In addition, when the DSS was in a situation where the EDD was less promising and timely information was requested then the DSS adapted by communicating less frequently. Although we only tested with specific EDD's, RCBE can in principle deal with any shape of the EDD.

We can conclude that the novel *integrated utility function* is a more direct and intuitive function for estimating the utility of probabilistic state estimates than the utility function we have used in previous articles and than information divergence measures, as discussed in chapter 2. Most importantly, the function is well able to judge the relevance of information.

We may also conclude that it in time-critical operations and resource constrained networks, standard communication protocols that provide transmission schedules can be enhanced by the ability to reevaluate information before every transmission attempt. In other words, it is worthwhile to combine communication protocols with RCBE.

In this chapter we have applied RCBE to enable run-time adaptive communication on a single information abstraction level. Because multiple entities act on different levels of the information abstraction hierarchy, it is likely that they simultaneously want to communicate information. In the future we would like to investigate how RCBE can run-time determine at which information abstraction level it is most rewarding to communicate. For example, some situations in maritime operations, as described in the introduction, could indeed benefit from communicating tracks in contrast to single measurements. Other situations can require the transmission of single measurements.

We have utilized a rather complex communication model, where multiple parameters influence the shape of both the EDD and the cost function. We have just begun exploring the influence of these parameters on the EDD and cost function and a we would like to do a more thorough exploration in the future.

In this investigation we did experiments on information-requests that regard timeliness and accuracy of probabilistic state estimates. It would be really interesting to experiment on more types of information-requests, *e.g.* requests regarding classification and investigate how RCBE could deal with information-requests regarding *ambiguity* of information.

We have tested on maritime scenarios using radars and would like to show that RCBE can also work in other application domains like mobility and security. Another domain is research regarding the issue of platooning in traffic, where the cars in a platoon need to maintain run-time ISA to react run-time to the dynamics within in the platoon.

Another application area where adaptive creation of ISA could be useful is in robotics. A team of robots can optimize their plans and actions given ISA. Based on the current information-requests individual robots can decide which sensory information to communicate.

The management system chooses sensors and allocates tasks for these sensors to fulfill a certain information-request. We can translate this into our terminology by saying that there is a certain information-request for a certain situation awareness.

Lastly, RCBE could be applied to *sensor management*. RCBE could calculate the reward of a sensor by balancing the value of the sensors' information and the cost of the sensors' active collection (*i.e.* energy, time). For example, when a ship is operating near the coast and needs detailed and timely positional information of a certain object on land, the ship can calculate the *expected reward* of sending out an Unmanned Air Vehicle (UAV). The UAV has the task to fly to the region where the object is, to observe the object with its camera and to send the information back to the ship. The question is therefore whether the *expected value* of the information is worth the costs (i.e. energy for flying to the region, energy for communicating the information to the ship and the delay of flying and communication).

To conclude, RCBE enables the DSS to run-time adapt communication by realistically rewarding information based on the current constraints due to communication and the current information-requests. We believe that RCBE is applicable in multiple domains, applicable for multiple information-requests and highly variable communication situations, and applicable at multiple information abstraction levels.

Chapter 6

Adaptive Team Formation to Optimize Shared Awareness

6.1 Introduction

This chapter investigate Adaptive Team Formation—ATF—, which is partly based on material from [1]. In addition to the method Request and Constraint Based Evaluation—RCBE—, introduced in chapter 5, ATF adds another dimension of adaptivity for the Distributed Sensor System—DSS—and increases the effectiveness and efficiency of executing its current task. In this thesis the main task of the system is to construct and maintain *Identical Shared Awareness* for the objects in the visible environment. Where RCBE run-time determines whether each newly collected information about these objects is valuable enough to share with the other agents in team $\mathcal{S}_o$, ATF dynamically determines the team entities. In other words, RCBE determines whether to share and ATF determines whom to share with. Therefore, ATF has to determine what the entities' current *contribution* is to the team-objective concerning the perceived object o.

The contribution of an entity to the team objective for an object can be determined from:

- the current visibility of the object
- the current effectiveness on the object
- the current communication capabilities with the other entities
- the quality of its sensors
- the utility its information brings to the ISA

First, the contribution is dependent on whether the entity has the object in its *effector range* and in its *detection range*. For example, it can be that an entity can act

[1] (van Foeken and Kester [2012])

on an object but has no local visibility or that it can perceive the object but that its effectors cannot reach it.

Second, there are constraints in the communication channel and the communication system that influence the delay of transmission and therefore delay the synchronization of the ISA. In time-critical situations, where immediate reaction is required, it is important to have an ISA that is highly accurate and is constantly up-to-date. In such situations, when an entity has troublesome communication with the other entities and receiving and transmission of sensory information is delayed, this degrades the value that the information brings to the ISA.

Third, the quality of the sensory information itself is a measure of how it impacts the ISA and the effectiveness on the object. When the sensor itself has a low resolution, a low detection rate or a small range, it may not at all be able to deliver detections worthy of communication.

Last, part of achieving the team objective is to take action, and for optimizing these actions the ISA needs to be optimized. This means that certain features of the ISA are more important than others, depending on the current team objective. Therefore, the relevancy of an agent can be measured by how its information impacts the currently considered importance of the ISA. It can be that a certain sensor is very good, but it does not improve the quality of important features.

All these four assessments can be dynamic during the lifespan of an object, which can have dynamic effects on the contributions of entities. ATF has as goal that the DSS can adapt to these dynamic changes.

Evaluating this contribution is *relevant* in the leading examples of this thesis, which are maritime scenario's where the environment is observed by multiple ships. Most striking are examples where ships are threatened by fast moving incoming missiles. Such threats can only be dealt with when the ships have a sufficiently accurate estimate of the location, speed, direction and acceleration of the missiles.

To deal with either improving or degrading contributions of entities to the objective for an object, Adaptive Team Formation—ATF—is suggested, which periodically adapts the teams of agents for every object. The method evaluates the contribution, and determines which agents should remain, be excluded or be included in the team. It does this all based on its effector range, detection range, communication situation, sensing capabilities, and current objective. In this chapter we test this method and the research question is therefore:

> Does the Adaptive Team Formation help to improve the Identical Shared Awareness?

A team is defined as a set of agents, $\mathcal{S}_o$, that share an identical awareness of an object o. An agent, a_{oj}, is defined as an autonomous entity active on an entity j and in a certain team $\mathcal{S}_o$. To determine whether information is worth sharing, each agent uses the evaluation method of the previous chapter: Request and Constraint Based Evaluation (RCBE). This method determines whether sensor readings contribute enough to the current ISA to be shared.

Through the simulation experiments, the research question is answered by comparing the qualities of the ISAs in two cases: First, when the team remains fixed throughout the simulation runs, but that RCBE is used to have some means of adaptive communication. Second, when in addition to RCBE, ATF is used to adapt the teams to differing situations. In van Foeken and Kester [2009] it is shown that RCBE gives better results than random, so if the addition of ATF results in even better ISA's the use of the method is shown.

6.2 ATF method

In case of degrading or improving contributions of entities, it might be valuable to exclude or include agents respectively. Deciding to include or exclude an agent involves the problem of recognizing to what degree an agent influences the expected effect on an object. Clearly, a disadvantage of exclusion is that an excluded agent does not have an updated ISA, which disables it to *optimally* coordinate actions with the other entities. It also stops delivering its collected information to the other agents, which can result in worse qualities of the ISA, such as a less accurate estimated location, or delayed detection of objects because they enter the smaller detection range of the smaller team later. Advantages of exclusion may be that the delay of communication among the members of the new team decreases so that the ISA is synchronized sooner and reaction time is shortened and that the costs of communication decrease. Moreover, if the excluded agent has the object within visual range, it logically has *no delay* in updating its local object awareness—LA, and can therefore affect the object *directly*.

Linking new agents in the team has the advantage that there is a greater area-coverage, robustness and that the ISA is fed with additional object-information so that, for example, accuracy increases. The additional entities on which the new agents reside also may increase the effectiveness on the objects, because first, they can work together instead of alone and, second, the joint effectors have exponentially more impact. The latter can be exemplified as follows: Two ships can eliminate a target fully at once with their joint effectors, while one ship has to hit the target twice. These are all advantages of linking new agents. The downside of linking, however, is that the costs of communication will increase and the delay of communication, synchronization and reaction will lengthen.

In short, the decision of including or excluding is based on the difference between the *expected reward* of including an entity and the *expected reward* of excluding the entity. The expected reward being a trade-off between the utility of the awareness and the communication costs and delay.

The agents have knowledge about:

- the communication situation with the other entities,
- their sensors,
- their local tracking algorithm,
- the current information-requests,
- the tracks they have within their detection range.

The ATF utilizes this knowledge to calculate the expected reward that an agent can bring to the ISA.

6.2.1 Set Up

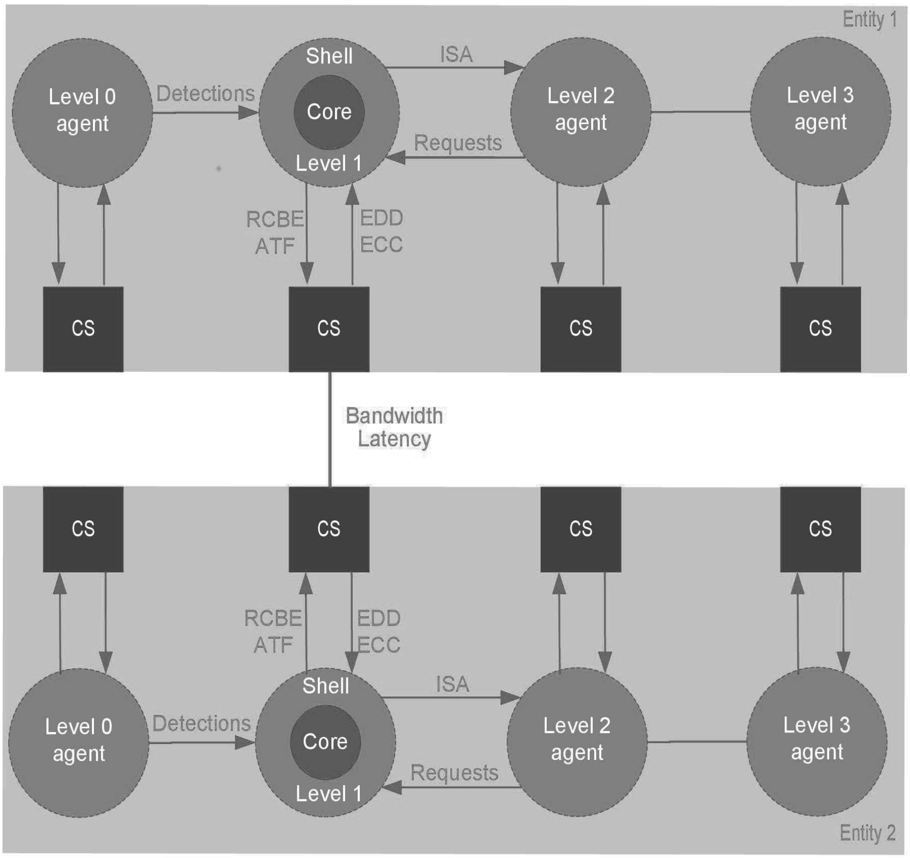

Figure 6.1:

Before describing the details of ATF, the general set-up of the DSS, as done in 2.9, is repeated shortly here. The schematic layout of the system is repeated as well in Fig. 6.1.

An agent a_{oj} is active on an entity j and maintains an ISA, $\hat{x}_o$, with other agents in team $\mathcal{S}_o$. The agent uses a kalman filter ϕ that fuses level 0 detections, $\mathcal{Z}_o = [z_1, z_2, z_3, ...]$ into level 1 updated state estimates, $\hat{x}_o \mid \mathcal{Z}_o = \phi(\hat{x}_o, \mathcal{Z}_o)$. Surrounding the core is the *Shell*, which harbors the evaluation methods RCBE, ψ, and ATF, φ. The *Shell* is able to transmit and can receive, via the CS, level 0 detections and level 1 tracks. It also can request Expected Delay Distributions

(EDDs) as discussed in chapter 3 eq 3.21:

$$\mathrm{EDD} = \hat{p}\left(t, \mathcal{Z}, \mathcal{S}\right) = \frac{\mathrm{d}(\hat{\mathcal{P}}\left(t, \mathcal{Z}, \mathcal{S}\right)}{\mathrm{d}t} \tag{6.1}$$

and Expected Communication Costs (ECCs) eq 3.23:

$$\mathrm{ECC} = \hat{\mathcal{C}}(t, \mathcal{Z}, \mathcal{S}) = \sum_{n=1}^{N} \hat{c}(n, \mathcal{Z}, \mathcal{S}) \tag{6.2}$$

from the CS. The agent receives updates of the information-requests, $U_{\mathcal{S}}(\Gamma(\hat{x}))$, from the higher-level agent. See Fig. 6.2 for an illustration of level 1.

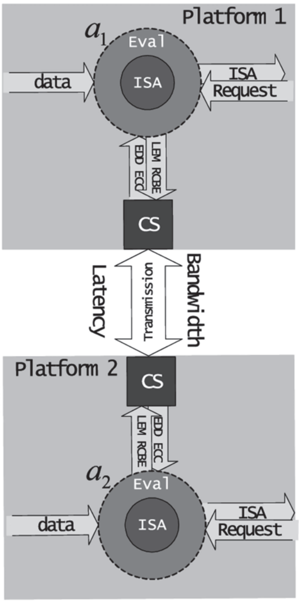

Figure 6.2: A generic layout of the important features of our distributed sensor system.
This minimal network consists of two interacting entities where communication resources are constrained. Two level 1 agents deliver requested level 1 information to the higher level. In order to deliver information they collect level 0 data from lower level agents and possibly each-other. At the heart of the agents lies the core functionality. The core is encircled by a *Shell* that functions as a mechanism that can adapt to dynamically changing information-requests and dynamically changing communication/processing or memory constraints.

Adaptive Team Formation—ATF—is a method used for every object that an entity can observe, affect or both. It acts at run-time, and is triggered by different events. Its task is to determine for every perceived object whether its team remains constant, continues with a new member or continues with a member less. Depending on the type of evaluation that ATF needs to do it requires a subset of the following information:

- $\hat{x}_o$ the current shared identical state estimate—ISA—of the perceived object o,
- $\hat{y}_o$ the current local state estimate of the perceived object o,
- $\hat{x}_{ro}$ the current reference state estimate of the perceived object o. This is best estimated state possible and has as goal to compared with another estimated state in order to conceive a utility. See section 4.2,
- $\mathcal{S}_o$ the current object-team,
- $\hat{p}$ the current expected delay distributions,
- $\hat{\mathcal{C}}$ the current ECCs,
- U the utility functions

The team, $\mathcal{S}_o$, shares an identical state estimate—that is an ISA, $\hat{x}_o$, of the perceived object, o. There is a leader of the team, $a_{oj} \in \mathcal{S}_o$, who performs certain later mentioned parts of the ATF. Such a team can be very transitory, whose lifetime is limited by several constraints, such as object-visibility duration or the duration of interest for the object. Next to keeping an ISA, each team-member with object-visibility maintains a local state estimate, $\hat{y}_o$, and a reference state estimate, $\hat{x}_{ro}$ of the object.

In general, ATF is defined by the following function: $\varphi(\hat{x}_o, \mathcal{Z}_o, \mathcal{S}_o) : \mathbb{R} \to \{0, 1\}$. It determines whether agent a_{oj} should not or should be in team $\mathcal{S}_o$—hence the boolean output $\{0, 1\}$, based on the contribution the information $\mathcal{Z}_o$ brings to the shared state estimate $\hat{x}_o$. ATF can be evoked when:

1. a new local track, $\hat{y}_o$, or an already shared track, $\hat{x}_o$, can be observed or can be acted upon by another entity.
2. a certain time-interval has passed after the last time an ATF was done.

If a new local track, $\hat{y}_o$ is found of object o by entity j, the object will be represented by a new agent, a_{oj}, and a new team, $\mathcal{S}_o$. When this object comes in either the effector, detection or both ranges of another entity, agent a_{oj} starts the ATF process for inclusion. If awareness of an object is already being shared, $\hat{x}_o$, and the object is in or enters either or both ranges of another agent, the leader of the team, a_{oj}, will start the ATF process for inclusion.

The moment that a single agent starts sharing its local object-awareness with another agent, a time-variable, m_o, is set to indicate when ATF has last been run. Another variable, n, determines the interval, $[0, n]$, after which ATF has to be run again. This interval is chosen such that it is significantly longer than the interval between two successive detections. m is reset when ATF has been run. At time $m_o + n$, each agent within $\mathcal{S}_o$ will perform ATF for exclusion.

Fig. 6.3 shows a diagram that illustrates the process of ATF.

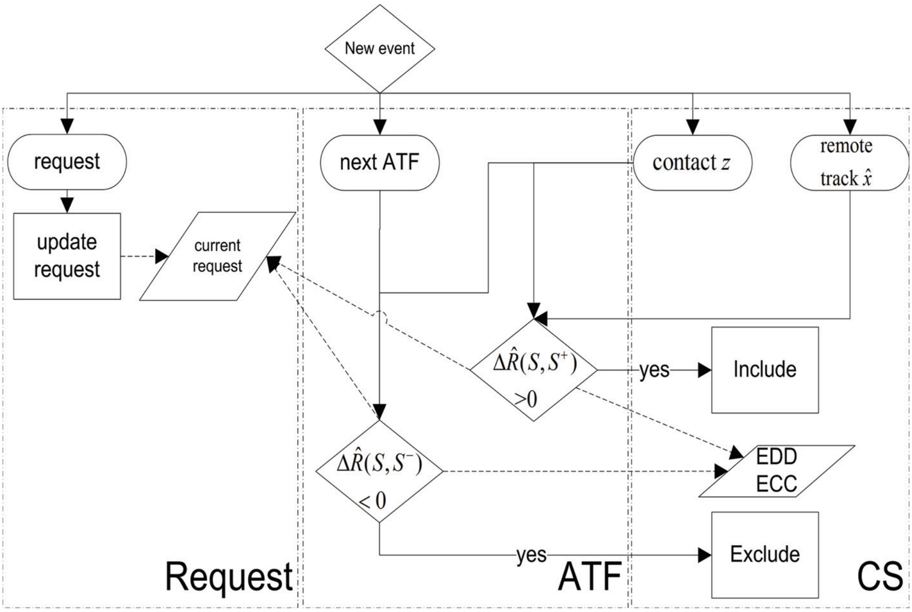

Figure 6.3: Flowchart showing the steps for ATF. Event 1: request update. Event 2: n ms have passed; it is time for a new ATF. Event 3: entity receives new contact z. Event 4: entity receives a remote track $\hat{x}$. If the entity is up for inclusion after a new contact has been received it uses current EDD and ECC and request to determine if inclusion is more rewarding than exclusion. If it is the moment for possible exclusion it calculates if exclusion would be more rewarding than staying. If so, it is excluded.

Utility

Determining $\varphi(\hat{x}_o, \mathcal{Z}_o, \mathcal{S}_o)$ involves calculating the expected utility of $\hat{x}_o$ for the team including and excluding the considered agent a_j. Recall from Ch. 4 the formula f

$$f\left(U_1, ..., U_{|\mathcal{S}|}\right) = |\mathcal{S}|^{\alpha} \sum_{a=1}^{|\mathcal{S}|} U_a, \tag{4.3}$$

for defining the utility function of a team, $U_{\mathcal{S}}$. The parameter α 4.3 determines the a-priori profit of cooperation in a team. If $\alpha > 0$ there is a profit to cooperate, as the team utility is larger than the sum of the local utilities. Also the probability of cooperation increases with increasing α. If $\alpha = 0$ the team utility is the sum of the local utilities and there is no profit of cooperation. If $\alpha < 0$ there is a punishment for cooperation.

Inclusion

The first time an object o is detected by one of the entities, $j \in \mathcal{J}$, the object-assessment agent, a_{jo}, on the entity will start a *new track*, $\hat{y}_o$. Later, another entity could be able to observe or affect the object. At such a moment the question is whether object-participation affects the object more than when they would work individually. If so, team $\mathcal{S}_o$ is born. Also, when the object has been tracked by multiple agents and enters the detection or effector range of a new entity, the same choice has to be made: should that new entity be included in the current team $\mathcal{S}_o$ to become $\mathcal{S}_o^+$?

There are three possible cases where an entity j is up for inclusion, which all require different calculations:

Case 1 The object only enters the effector range

Case 2 The object only enters the detection range

Case 3 The object enters both effector and detection range

In the *first case* the entity cannot detect the object. This means the entity does not have any local information that can be used to calculate the reward. This simplifies the process significantly since any agent that is already in team $\mathcal{S}_o$ has all the information to determine whether the entity should be included or not. Such an agent simply calculates: $\Delta\hat{\mathcal{R}}(\mathcal{S}_o, \mathcal{S}_o^+) = \left(\hat{\mathcal{R}}_{\text{inc}}\left(\hat{x}_o, \hat{x}_o, \mathcal{S}_o^+\right) - \hat{\mathcal{R}}_{\text{exc}}\left(\hat{x}_o, \emptyset, \mathcal{S}\right)\right)$ and contributes to the expected change of reward when entity j will participate. The three parameters in $\hat{\mathcal{R}}$ are:

1. The information of which the reward is calculated. In this case the ISA $\hat{x}_o$.
2. The information to be send. In the case of sharing the total ISA is to be communicated.
3. The team for which the reward is calculated.

That entity has to determine whether it is worth to share track state $\hat{x}_o$ with entity j. The first term is defined as:

$$\hat{\mathcal{R}}_{\text{inc}}\left(\hat{x}_o, \hat{x}_o, \mathcal{S}_o^+\right) = \int_{t_0}^{t_{\max}} \hat{p}\left(t, \hat{x}_o, \mathcal{S}_o^+\right) \left[\mathcal{U}_{\mathcal{S}_o^+}\left(\epsilon\left(\hat{x}_o\left(t\right)\right)\right) - \hat{\mathcal{C}}\left(t, \hat{x}_o, \mathcal{S}_o^+\right)\right] \mathrm{d}t - \hat{q}\hat{\mathcal{C}}\left(t_{\max}, \hat{x}_o, \mathcal{S}_o^+\right) \tag{6.3}$$

This calculation involves the utility that the larger team is expected to have and the costs that are expected to be made to communicate the $\hat{x}_o$. In detail it involves integrating over the delay t until $t_{\max}$. $t_{\max}$ is reached at the moment $\mathcal{U}_{\mathcal{S}_o^+}\left(\epsilon\left(\hat{x}_o\left(t\right)\right)\right) - \hat{\mathcal{C}}\left(t, \hat{x}_o, \mathcal{S}_o^+\right) <= 0$. In other words, when the expected costs of communication are higher than the expected utility of sharing. For each t the expected probability of successful communication, $\hat{p}$, is multiplied with the utility minus the costs. However, there is a probability, $\hat{q}$, that communication fails to a moment, $t_{\max}$, where the cost becomes higher than the value such that it would not be rewarding anymore to share $\hat{x}_o$. This also presents the probability that costs have been made during $[0, t_{\max}]$. Therefore the last term, $\hat{q}\hat{\mathcal{C}}\left(t_{\max}, \hat{x}_o, \mathcal{S}_o^+\right)$, is subtracted from the integral.

The expected reward of not sharing $\hat{x}_o$ is:

$$\hat{\mathcal{R}}_{\text{exc}}\left(\hat{x}_o, \emptyset, \mathcal{S}_o\right) = \mathcal{U}_{\mathcal{S}_o}\left(\epsilon\left(\hat{x}_o\left(t_0\right)\right)\right). \tag{6.4}$$

This simply calculates the utility of $\hat{x}_o$ for the team without agent a_o without delay. Then, if $\Delta\hat{\mathcal{R}}(\mathcal{S}_j, \mathcal{S}_j^+) > 0$, the agent sends track state $\hat{x}_j$ to entity j. Receiving this track gives birth to agent a_{oj} Fig. 6.3 shows that receiving this also initiates the CS to *include* itself in team $\mathcal{S}_o$.

In the second and the third case entity j has visibility of the object. The problem is that the entity has no history of detections associating with the track and is therefore unable to determine what its contribution would have been in the previous time-window. It can only determine what the impact of future detections will be. For this it first needs the track and therefore receives this from an agent in team $a_{oj} \in \mathcal{S}_o$. At reception the entity temporarily becomes part of the team so that it receives detections from the team and can send its own positively rewarded detections to the team. This to make the evaluation as accurate as possible.

We define a parameter g that determines the number of detections the entity will be using for evaluation. The higher g, the more accurate the evaluation. The parameter g must be larger than 1 and smaller than the number of detections done within interval n. Surely, a single detection is little information to base inclusion on, and therefore this parameter is used. The generic formula for both second and third case is:

$$\Delta\hat{\mathcal{R}}\left(\mathcal{S}_o, \mathcal{S}_o^+\right) = \sum_{z=z_1}^{z_g} \hat{\mathcal{R}}_{\text{inc}}\left(\hat{x}_o, z, \mathcal{S}_o^+\right) - \hat{\mathcal{R}}_{\text{exc}}\left(\hat{x}_o, \emptyset, \mathcal{S}_o\right) - \hat{\mathcal{R}}_{\text{exc}}\left(\hat{x}_o, z, j\right). \tag{6.5}$$

Where the first term defines the expected reward when sharing the detections, the second term the reward when not sharing the detections and the third term the reward of the detections for local track state, relevant for the local effector range of the object. Even though the generic formulation is the same in the second and third case, the calculation of its first and third term is different.

In the *second case*, the entity j has no local utility because it does not have an effector range on the object. Therefore utility is only calculated for the current team $\mathcal{S}_j$ and the first term for a single detection is:

$$\hat{\mathcal{R}}_{\text{inc}}\left(\hat{x}_o, z, \mathcal{S}_o^+\right) = \int_{t_0}^{t_{\max}} \hat{p}\left(t, z, \mathcal{S}_o^+\right)\left[\mathcal{U}_{\mathcal{S}_o}\left(\epsilon\left(\hat{x}\left(t\right)\right)\right) - \hat{\mathcal{C}}\left(t, z, \mathcal{S}_o^+\right)\right]\mathrm{d}t - \hat{q}\hat{\mathcal{C}}\left(t_{\max}, z, \mathcal{S}_o^+\right) \tag{6.6}$$

The second term is the expected reward of the state without the detection for the current team $\mathcal{S}_o$: $\hat{\mathcal{R}}_{\text{exc}}\left(\hat{x}_o, \emptyset, \mathcal{S}_j\right) = \mathcal{U}_{\mathcal{S}_j}\left(\epsilon\left(\hat{x}\left(t_0\right)\right)\right)$. And as there is no local utility, the third term is zero: $\hat{\mathcal{R}}_{\text{exc}}\left(\hat{x}_o, z_o, a\right) = 0$.

In the *third case*, the agent has communication delay and cost as well as utility for detection z. The first term is almost equal to (6.6) with the exception of using the utility for the possible new team, $\mathcal{U}_{\mathcal{S}_o^+}$:

$$\hat{\mathcal{R}}_{\text{inc}}\left(\hat{x}_o, z, \mathcal{S}_o^+\right) = \int_{t_0}^{t_{\max}} \hat{p}\left(t, z, \mathcal{S}_o^+\right) \left[\mathcal{U}_{\mathcal{S}_o^+}\left(\epsilon\left(\hat{x}_o\left(t\right)\right)\right) - \hat{\mathcal{C}}\left(t, z, \mathcal{S}_o^+\right)\right] \mathrm{d}t - q\hat{\mathcal{C}}\left(t_{\max}, z, \mathcal{S}_o^+\right) \tag{6.7}$$

The second term, the expected reward for not sharing, is again $\hat{\mathcal{R}}_{\text{exc}}\left(\hat{x}_o, \emptyset, \mathcal{S}_o\right) = \mathcal{U}_{\mathcal{S}_o}\left(\epsilon\left(\hat{x}_o\left(t_0\right)\right)\right)$. But, as there is local utility the third term, the summed local utility of the local track state, is not zero: $\hat{\mathcal{R}}_{\text{exc}}\left(\hat{x}_o, z, j\right) = \mathcal{U}_o\left(\epsilon\left(\hat{x}_o\left(t\right)\right)\right)$.

Exclusion

If there has not been any ATF moment for n seconds it is time for the *next ATF*, see figure Fig. 6.3, and all the agents, a_{oj}, comprising all the current teams, $\mathcal{S}_o$, calculate whether they should remain or be excluded from their team. For a single agent a_{oj} the function is therefore defined as the expected change of reward when the team would have acted without that agent instead:

$$\varphi(\mathcal{S}_o, \hat{x}_o, \mathcal{Z}_o) \equiv \Delta\hat{\mathcal{R}}\left(\mathcal{S}_o, \mathcal{S}_o^-\right) >= 0 \rightarrow \mathcal{S}_o^- : \mathcal{S}_o \tag{6.8}$$

If $\Delta\hat{\mathcal{R}}$ is positive, it would probably have been better to work without a and results in the new team being $\mathcal{S}_o^-$ indicated by $\rightarrow \mathcal{S}_o^-$, if negative the agent remains $\mathcal{S}_o$, indicated by $: \mathcal{S}_o$. The difference in reward is calculated as follows:

$$\Delta\hat{\mathcal{R}}(\mathcal{S}_o, \mathcal{S}_o^-) = \hat{\mathcal{R}}_{\text{exc}}\left(\hat{x}_o, \emptyset, \mathcal{S}_o^-\right) + \hat{\mathcal{R}}_{\text{exc}}\left(\hat{y}_o, \mathcal{Z}_o, a_{oj}\right) - \hat{\mathcal{R}}_{\text{inc}}\left(\hat{x}, \mathcal{Z}_o, \mathcal{S}_o\right) \tag{6.9}$$

The first and second term connote the expected reward when agent a_{oj} would have been excluded from the team—that is the reward of state $\hat{x}_o$ for $\mathcal{S}^-$ without $\mathcal{Z}_o$ plus the reward that $\mathcal{Z}_o$ would have brought to agent a_{oj} locally, hence without sharing. The third term of (6.9) signifies the expected reward that the previous detections $\mathcal{Z}_o$, would have brought to the shared track state $\hat{x}_o$. $t_0 : t = 0$

The first and second term are broken down as follows:

$$\hat{\mathcal{R}}_{\text{exc}}\left(\hat{x}_o, \emptyset, \mathcal{S}_o^-\right) = \sum_{z \in \mathcal{Z}_o} \left[\mathcal{U}_{\mathcal{S}_o^-}\left(\epsilon\left(\hat{x}_o\left(t_0\right)\right)\right)\right] \tag{6.10}$$

$$\hat{\mathcal{R}}_{\text{exc}}\left(\hat{y}_o, \mathcal{Z}_o, a_{oj}\right) = \sum_{z \in \mathcal{Z}_o} \left[\mathcal{U}_a\left(\epsilon\left(\hat{y}_o\left(t_0\right)\right)\right)\right] \tag{6.11}$$

In both cases in the calculation there is no communication, so there is no delay and costs. The first term is the summed utilities of the current state at the times when detections $\mathcal{Z}_a$ were measured, but without the detections actually included. This simulates the case that a_{oj} has not shared its detections. The second term is the summed local utility of the local track state at those moments.

The third term is the reward when the agent a_{oj} is incorporated in the team and given by:

$$\hat{\mathcal{R}}_{\text{inc}}\left(\hat{x}_o, \mathcal{Z}_o, \mathcal{S}_o\right) = \sum_{z \in \mathcal{Z}_o} \int_{t_0}^{t_{\max}} \hat{p}\left(t, z, \mathcal{S}_o\right) \left[\mathcal{U}_{\mathcal{S}_o}\left(\epsilon\left(\hat{x}_o\left(t\right)\right)\right) - \hat{\mathcal{C}}\left(t, z, \mathcal{S}_o\right)\right] \mathrm{d}t - \hat{q}\hat{\mathcal{C}}\left(t_{\max}, z, \mathcal{S}_o\right) \quad (6.12)$$

As Fig. 6.3 shows, the *current request*, all previously local contacts that associated with $\hat{x}$ between $t - m$ and t that is set $\mathcal{Z}_o$, and the EDDs and ECCs are needed for this calculation. For every $z \in \mathcal{Z}_o$ the expected reward is calculated and summed. The expected reward of a detection consists of two terms. The first is an integral over the delay interval $[0, t_{\max}]$. For every t the difference between utility of z for state $\hat{x}_o$ and cost of communication is multiplied with the expected success probability of sharing z, $\hat{p}$. However, there is a probability, $\hat{q}$, that communication fails to a moment, $t_{\max}$, and therefore the second term is subtracted from the integral.

In conclusion, the loss or the gain of reward by excluding agent a_{oj} is estimated by comparing the effectiveness on the object of agent a_{oj} including in $\mathcal{S}_o$ with that of $\mathcal{S}_o^-$ and agent a_{oj} working separately. The combined utility of $\mathcal{S}_o$ is, as a rule, higher for the same observed state $\hat{x}_o$, but the delay may deteriorate the state so severely or the costs or communication may be so high, that the estimated reward is equal or lower. In that event the agent is excluded from the team $\mathcal{S}_o$.

When an agent did not have any visibility or effector range on the object during the previous time-window the two terms will be equal: $\hat{\mathcal{R}}_{\text{exc}} == \hat{\mathcal{R}}_{inc}$ and therefore it will be excluded. First of all, the agent did not collect any detections associated with the track, which makes the costs zero and obviously causes no delay. Therefore $\hat{\mathcal{R}}_{\text{inc}} = \mathcal{U}_{\mathcal{S}_o}\left(\epsilon\left(\hat{x}_o\left(t_0\right)\right)\right)$. Then, since the state lies out of its effector range, the shared awareness does not have an added utility for the agent. This means that $\mathcal{U}_{\mathcal{S}_o}\left(\epsilon\left(\hat{x}_o\left(t_0\right)\right)\right) == \mathcal{U}_{\mathcal{S}_o^-}\left(\epsilon\left(\hat{x}_o\left(t_0\right)\right)\right)$. Locally the agent does not have an awareness of the object and makes the local utility logically $0, \mathcal{U}_a\left(\epsilon\left(\hat{y}_o\left(t_0\right)\right)\right) == 0$. In conclusion: $\hat{\mathcal{R}}_{\text{exc}}\left(\hat{x}_o, \emptyset, \mathcal{S}_o^-\right) = \mathcal{U}_{\mathcal{S}_o^-}\left(\epsilon\left(\hat{x}_o\left(t_0\right)\right)\right) = \hat{\mathcal{R}}_{\text{inc}} = \mathcal{U}_{\mathcal{S}_o}\left(\epsilon\left(\hat{x}_o\left(t_0\right)\right)\right)$

6.2.2 Performance measures

In 2.7 we claimed that performance measures should relate directly to the goals of the system. In this chapter performance of the DSS is measured in terms of *offline reward*. Instead of measuring the Mean Squared Error of the resulting tracks compared to the ground truth, like we did with the experiments regarding RCBE, the resulting tracks are measured in how much utility the tracks have brought in relation to the currently relevant features of information and how much communication costs have been made. We measure the offline reward—utility minus costs—at the moments that measurements have been made.

The offline reward can be formalized as:

$$\mathcal{R}\left(\hat{x}_o, \mathcal{Z}, \mathcal{S}_o\right) = \sum_{z \in \mathcal{Z}_o} \mathcal{U}_{\mathcal{S}_o}\left(\epsilon\left(\hat{x}_o\left(t_{\text{real}}\right)\right)\right) - \mathcal{C}\left(t_{\text{real}}, z_o, \mathcal{S}_o\right), \quad (6.13)$$

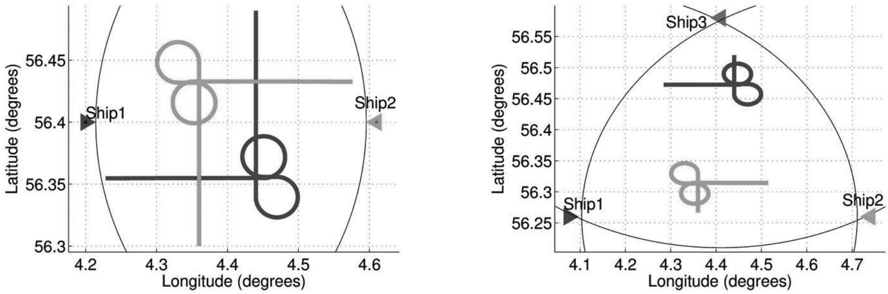

Figure 6.4: (Left) A snapshot of the first scenario
Two ships with the black circled (overlapping) detection ranges observe two flying objects. The routes of the objects are shown in blue and green. (Right) A snapshot of the second scenario. Virtually the same as the first scenario, but extended with one extra ship observing the area.

where t_{real} represents the delay for incorporating the detection. This expresses the utility of the ISA $\hat{x}$ at the moment that the detection z was incorporated in the ISA minus the costs that were made to communicate the detection. As RCBE is used, some detections are not shared. In this case there are no costs and there is no delay resulting in solely measuring the utility:

$$\mathcal{R}\left(\hat{x}_o, z, \mathcal{S}_o\right) = \mathcal{U}_{\mathcal{S}_o}\left(\epsilon\left(\hat{x}_o\left(t_{\text{real}} = 0\right)\right)\right) \tag{6.14}$$

This equation is also used when the team consists of only one agent, hence measuring the reward of the local tracks.

6.3 Experimental Results

Fig. 6.4 shows the two scenarios of a DSS. A realistic simulation environment as described in 2.8, the system of van Iersel et al. [2008], is used to run the experiments. Scenario 1 comprised two ships j_1, j_2 and scenario 2 three ships j_1, j_2, j_3. The ships used radars to keep objects in their visual range under surveillance and the level 1 agents together maintained an ISA of the tracks of the objects in the area. Level 0 agents served as a local source of noisy detections. A single unidentified object, which was in truth a hostile fighter, flew first into the detection and effector range of ship 1 and later into the range of ship 2 as well. In the second scenario the object flew first in the detection and effector range of ship 1 then in the ranges of ship 3 and finally in the ones of ship 2. Ship 3 is placed in between ship 1 and 2 and the object moves in a straight line.

These scenarios represent a DSS in need of timely and accurate information about the objects in the environment in a communication constrained network. The ships needed to surveil the environment for hostile fighters. However, frequently transmitting contacts with limited communication resources, result in high communication costs and significant delays. Delays decrease the accuracy of the ISA,

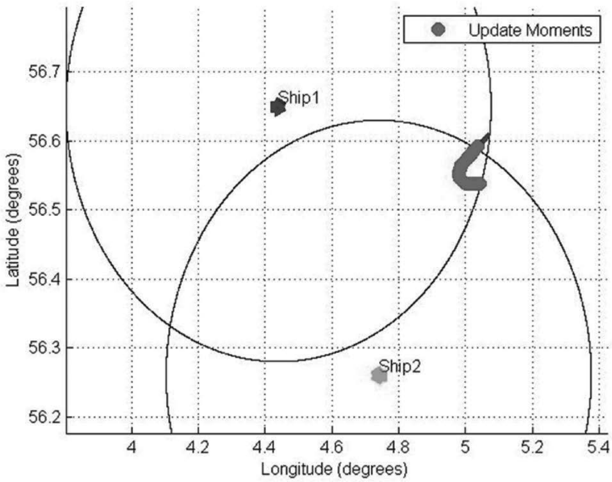

Figure 6.5: A snapshot of the first scenario. Two ships with the black circled (overlapping) detection ranges observe a single flying object. The routes of object is shown in blue where the red rots express the update moments by the tracker.

hence the effectiveness of surveillance task. In conclusion, these scenarios are suited for testing to what extent ATF can maintain a high quality ISA.

We measure the offline reward—utility minus costs— see (6.13) of the resulting tracks at the moments that measurements have been made.

The ships get the utility function of Fig .6.6 as an information-request. This utility function is given in Fig .6.6 and is the same as was used in the experiments of chapter 5.

Furthermore, there are some important parameters to be set:

g The number of detections used to determine whether a new entity should be included is 2.

n The interval after which there is a new exclusion moment is 10 s. Each entity generates contacts every second. This means 10 contacts per interval are used.

6.3.1 Results scenario 1

We have run the scenario with three different settings:

1. $\alpha = 0$ and using only RCBE.
2. $\alpha = 0$ and using ATF as well as RCBE.
3. $\alpha = 0.5$ and using ATF as well as RCBE.

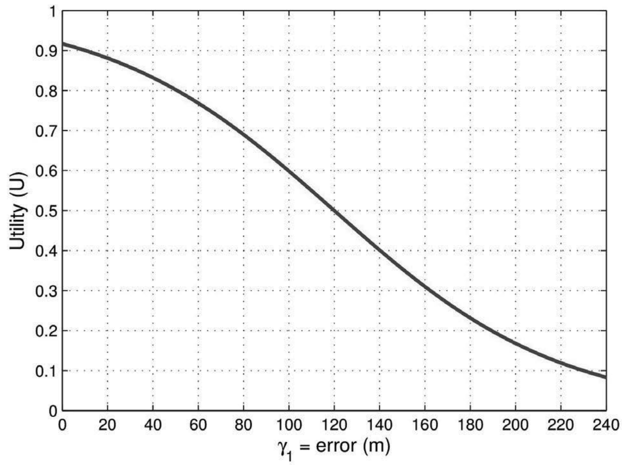

Figure 6.6: The accuracy request of agents a_2 and a_2 take the form of this utility function.

The parameter α determines the a-priori profit of cooperation in a team. When $\alpha = 0$ there is no advantage of cooperation (or ISA). g is 2 and n is 10 seconds. What we would like to show is that ATF rightly chooses not to cooperate in case $\alpha = 0$ because the sum of local rewards will by definition always be higher than the shared reward. We also would like to show that ATF rightly chooses to cooperate in case $\alpha = 0.5$.

Fig. 6.7 plots the offline rewards of:

1. the local tracks of the object when using ATF with $\alpha = 0$. Over time there is no cooperation between the ships since ATF chooses to do so, hence the ships do not have a shared track. The offline reward is simply the same as the offline utility of the track error in relation to the ground truth—blue and red

2. the sum of the offline rewards of the local tracks—green

3. the shared track when only using RCBE with $\alpha = 0$—blue green

4. the shared track when using ATF when $\alpha = 0.5$—purple

What can be seen is that the reward of the track when only using RCBE is equal or lower than the sum of the local rewards. This is logical since $\alpha = 0$ and because less detections are used for the shared track—some detections are not rewarding enough for communication. Moreover, the delay and costs of communication also influence the offline reward negatively.

Taking these aspects into account and the fact that communication costs are made makes it clear that in this situation ATF has merits.

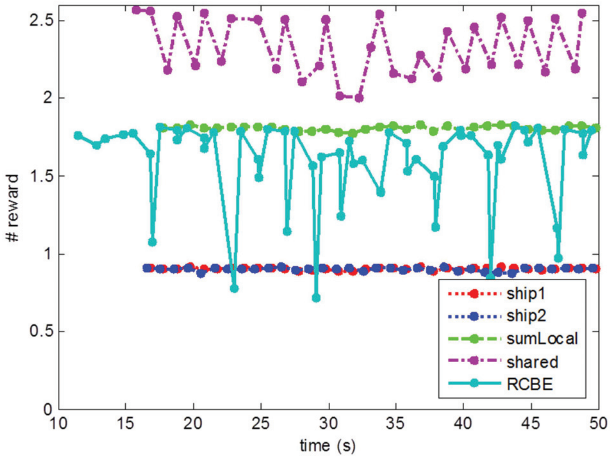

Figure 6.7: The offline rewards of the local estimates in blue and red, of the sum of the two local rewards in green, of when using only RCBE in light blue and in purple when ATF is used with an $\alpha = 0.5$.

In the next experiment α is raised to 0.5 to increase the effect of cooperation. This to show that ATF does choose to cooperate when the sum of local expected rewards is always lower than the shared reward. The purple line shows this offline reward of the shared track. It can be seen that ATF has done well to choose cooperation since the shared reward always exceeds the sum of the local offline rewards. Apparently this is enough to compensate for the costs of communication.

6.3.2 Results scenario 2

We have run the scenario with three different settings:

1. $\alpha = 0.4$ and using only RCBE.
2. $\alpha = 0.4$ and using ATF as well as RCBE.
3. $\alpha = 0.7$ and using ATF as well as RCBE.

g is 2 and n is 10 seconds.

The results of the first two settings are displayed in Fig. 6.8. There the offline rewards are plotted of:

1. the local track by ship 2 of the object with—red,
2. the shared track of the object between ship 1 and 3—blue,
3. the sum of the previous two—green,

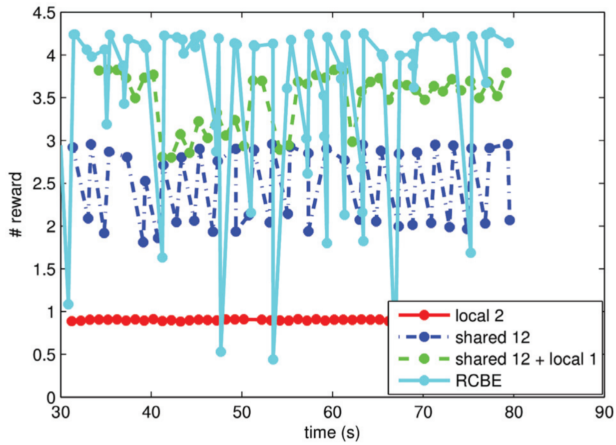

Figure 6.8: Case $\alpha = 0.4$: The offline reward of the local estimate of ship2 in red, of the shared estimate between ship 1 and 3 in blue, of when using only RCBE in light blue and green the sum of the first two

4. the shared track when only using RCBE—light blue,

In case $\alpha = 0.4$, ship 1 and ship 3 are cooperating in the ATF, but ship 2 is not. This can be explained as follows: The object moved into the effector and detection range of ship 1 first and then moved into the ranges of ship 3. At that point in time ship 3 received the track from ship 1 and evaluated itself to be rewarding enough to cooperate with ship 1. Hence they became a team: $\mathcal{S}_o = j_1, j_3$. When the object moved into the ranges of ship 2, ship 2 received the track from ship 1, since that is the leader of team $\mathcal{S}_o$. It evaluated itself to be negatively influencing the accuracy of the track, hence not including itself in the team. Observing the offline rewards, this did make sense. The light blue line represents the offline reward when only using RCBE and not ATF. This means that all ships are team members by default. The green line represents the offline reward when ATF was used, which is the sum of the blue and red line. The green line does not exceed the blue line all the time, but the valleys are so deep and below the lowest parts of the green line that the average offline reward is lower. These valleys indicate that the communication costs where really high. We can conclude ATF was useful, since it determined cooperation between three ships would not be rewarding due to the high communication costs.

If $\alpha = 0.7$ ship 2 did decide to cooperate, see Fig. 6.9. The offline reward of cooperation exceeded the green line, representing the sum of the offline reward of cooperation between 1 and 3 and the local offline reward of ship 2.

In conclusion, ATF is able to adapt the team formation to the degree that cooperation is appreciated (this is indicated by α) and to the communication costs.

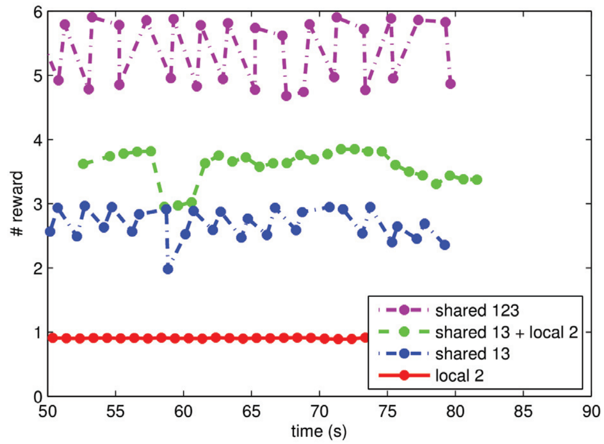

Figure 6.9: Case $\alpha = 0.7$: The offline reward of the local estimate of ship2 in red, of the shared estimate between ship 1 and 3 in blue of the sum of the previous two and the shared estimate between ship 1, 2 and three in purple

6.4 Discussion

In the two experiments we have attempted to answer the research question:

> Does the Adaptive Team Formations help to improve the quality of the Identical Shared Awareness?

In answering this question the results of the experiments were measured with a new performance measure, the *offline* reward presented in section 6.2.2. This performance measure gives a better evaluation of the experiments compared to the experiments in the previous chapter since it is able to directly relate the measure to the goals of the system instead of to the Mean Squared Error. When the goal of the system differs from the Mean Squared Error, like for example: *the detection must be as accurate such that my missile defense system can eliminate the the target with a 90 percent probability*, we put in a utility function representing exactly that goal. And we measure the performance after experimentation by calculating the offline reward with this specific utility function.

The first experiment showed the usefulness of ATF over RCBE. Cooperation was considered useless, by choosing an $\alpha = 0$, while in RCBE the entities always cooperate. When cooperation was considered useful to a certain degree, being that $\alpha = 0.5$, in this scenario ATF did choose to cooperate.

We argue that α is just one way of incorporating the action effectiveness into the utility function, and that there are other options. If there is more information about the effect of cooperation on the action effectiveness it can be worthwhile to put this effect into the utility. This effect can be expressed in certain features of information. As an example, a hostile object may have the same chance of being destroyed by two missiles with a lower location accuracy than by one missile with a higher location accuracy. Hence, if two ships cooperate a lower accuracy is needed to achieve the same utility than with a single ship. This information about the effect of cooperation can be modelled more directly in the utility function than with the α. One way of doing this is by making it one of the features Γ of information on which the utility function is based.

The parameters n, the interval between exclusion moments, and g, the number of detections which determine whether a new entity should be included or not were particularly chosen for the experiments in this chapter. However, these types of parameters can be extended to a broader scope of ATF observational tasks. It can be worthwile to investigate the influence of these parameters on the performance of the system. In particular because the optimal values of n and g may be different for different scenarios.. It can have advantages to have a lower n. Such as, that the DSS can adapt more quickly to changing situations. However, a downside is that there is less information to base the exclusion upon. The quality of the decision is influenced by $g <= n$. It determines how much information is used to make the exclusion decision. Of course, if there is offline knowledge about the correct setting of these parameters this can be used. Maybe, a team needs to be a team for a minimum amount of time to be effective. Also, it may be beneficial if these parameters are adaptive, or can be learned during a scenario.

The experiments comprised only two or three ships with a single object to track to show the usefulness of. In order to determine the scalability of ATF experiments with many more entities and multiple objects to track are needed.

6.5 Conclusions and Recommendations

The objective of this chapter was to show that Adaptive Team Formation (ATF) helps to improve the Identical Shared Awareness (ISA). ATF determines at runtime which entities should be part of the team to create shared awareness. Results are presented for creating ISA with two and with three ships. From these results it is shown that ATF, depending on the benefit of coordination of actions, automatically selects the proper entities in the team that should cooperate in tracking objects. It is an addition to Request and Constraint Based Evaluation (RCBE) since the results show that there are situations when cooperation results in lower reward than when no cooperation is present, or that a subset of all acting/observing entities is cooperating.

We introduced a novel measure for evaluating the system performance in relation to its goals: *the offline reward*. We have discussed the use of α as a parameter to incorporate a measure of action effectiveness of cooperation. We argued that α works as a way for the higher level to express the action effectiveness of cooperation but that incorporating α in the utility function as one of the features of information offers the higher level a more task specifical method.

In future experiments we would like to test the impact on team formation of varying communication capabilities. Moreover, we would like to research the scalability of ATF by having larger groups of entities and multiple objects to track.

In addition, we want to examine the adaptivity to run-time changes such as the communication capabilities or information-requests. Or the adaptivity to changing visibility, and changing effectiveness of an entity.

ATF is a framework that can be extended to other domains than tracking. It is interesting to investigate those domains.

Another interesting extension of ATF may be the ability of hopping information via entities to other entities. Hopping may upgrade the quality of communication and hence down the communication costs and delay. It may also help in case communication is faulty.

Chapter 7

Conclusions and future work

7.1 Conclusions

In this thesis we consider *Distributed Sensor Systems* maintaining an identical shared awareness of the environment with *limited communication resources*. The typical application of this thesis is a *maritime operation* with multiple ships, threatened by hostile objects. They can benefit from coordinating actions towards these objects. To enable coordination the ships need to share awareness of these objects.

The general objective of this thesis is:

> To provide innovative methods for creating the *best shared awareness* of objects in a dynamic environment within a communication constrained distributed sensor system with varying goals.

In such systems from a higher abstraction level *information-requests* are issued to a lower level team of agents. The team builds a *shared awareness* from information of all agents in the team which has to be communicated between them. However, this is often not completely possible due to constraints in communication between the entities where the agents are residing. Moreover, even when it is possible, transmission of information may be delayed too long. Therefore, we need at run-time a mechanism that can adapt to dynamically changing information-requests and dynamically changing communication/ processing or memory constraints. It simultaneously uses the information requests as well as the utility of the information and estimations of the communication situation to make decisions about *when* to communicate *what* to *whom*. For this decision making, the interaction with a communication shell is essential. At run-time, this shell is responsible for the actual transmission of information to other entities and can on request provide accurate and up-to-date communication information.

This results in the following objectives:

- Determine how to best relate the information-requests/utility functions of the higher information abstraction level to the information that the lower abstraction level produces,

- Simultaneously deal with varying information-requests and limited communication capabilities, chapter 5,
- Estimate the run-time communication capabilities, chapter 3,
- Determine the *utility* and *value* of information, chapter 4,
- Determine the contribution of entities to the *identical shared awareness*, chapters 4 and 5.

In chapter 2 we discussed the state-of-art in relation to our approach. The main goal of chapter 3 was to introduce a generic, low-complex communication model that serves as an enabler for our evaluation methods. We presented a novel low-complexity communication model of the system and the communication channel. The model enabled simulating accurate and up-to-date communication status information—in this case about the expected consumption of communication resources and expected latency. The most dominant performance indicators were identified and their relation to the underlying key parameters was modeled. A formal description of the *expected delay distribution*—EDD—was presented, followed by a formal description of the *expected cost of communication*—ECC. The simulation examples showed that EDDs and ECCs can be estimated for transmitting in different environmental circumstances and transmitting a message to different receiver groups. In addition the effect of reallocating resources on the EDD and ECC can be found. Evaluation methods can use this information to determine whether transmission is rewarding or to determine which entities should join a team for sharing awareness.

This model does not incorporate the most detailed parameters, but provides the key parameters to enable precise communication information for realistic simulations. Such low-complexity models are not present in the literature up till now. This model is novel in that it enables information evaluation methods to be adaptive to changing communication circumstances. You could also experimentally estimate the EDD and fit the parameters of your model. Moreover, the model enables the evaluation of the impact of different communication techniques on the EDD and ECC. And it also enables the evaluation of allocating certain parameters—i.e. resource management.

This model can be used to quickly compare the performance of different communication techniques in different scenarios. It can even be used to test the run-time adaptation of different communication techniques and decide which one to use. For example, some time-critical information can be sent by a low-latency technique-such as WiMAX, but other less time-critical information by a high-latency technique-such as Link 16.

The method for calculating the utility of information, the integrated utility function was introduced in chapter 4. It was shown how *information-requests* are formed in the system and how they are formalized as *utility functions*. Argumentation was given for the need of a *reference state* to calculate the utility of certain features, like the utility of the error of a track. In case of error the *reference state* is based on all the information that can be associated to the track. Subsequently, several information-theoretic methods for dealing with *information-requests* are discussed but it was concluded they are too indirect and implicit in their measuring of information-utility. New is that by combining the utility function, with certain evaluation parameters, we were able to capture the subjective importance of goals. This lead to the *integrated utility function* that can directly and precisely measure the utility of the important features of information. The last section presented the *value function* that calculates the gain in *expected utility* caused by new information. We based our approach on the method of Velagapudi et al. [2007] but make it a function of the relevant features related to the ISA instead of the agents local state estimate and replace their utility/cost function with the integrated utility function. Our goal is to construct ISA, where their method does not require this. We showed in an example the advantages of our approch. The *integrated utility function* and the *value function* are central in the functioning of the two evaluation methods presented in this thesis: *Request and Constraint Based Evaluation* (RCBE) and *Adaptive Team Formation* (ATF).

In chapter 5 we showed that RCBE enables distributed entities to maintain Identical Shared Awareness (ISA) and to run-time adapt their interaction to both the current information-requests and the current constraints due to communication. We have assumed that ISA is guaranteed when the transmitting entities receive acknowledgement messages from all the receiving entities. It follows that the DSS can perform coordinated actions based on this ISA.

We have done simulation experiments on two maritime scenarios, where multiple ships are instructed to maintain an ISA of the flying objects in their overlapping detection ranges. The results show that RCBE enables the ships to run-time adapt the communication to two different requests: a request for a certain *accuracy of tracks* and a multi-dimensional request for both a certain *accuracy of tracks* and *timeliness* of information. Otherwise stated, RCBE can cope with varying requests.

The results also show that RCBE simultaneously enables entities to adapt to varying levels of communication capabilities, *i.e.* varying data rates and varying bit-rate errors. The ISA improved significantly when the DSS used RCBE compared to the ISA when the DSS used the benchmark method. In addition, when the DSS was in a situation where the EDD was less promising and timely information was requested then the DSS adapted by communicating less frequently. Although we only tested with specific EDD's, we expect that RCBE can deal with any shape of the EDD.

We can conclude that the novel *integrated utility function* is a more direct and intuitive function for estimating the utility of probabilistic state estimates than the utility function we have used in previous articles van Foeken and Kester [2009] and than information divergence measures. Most importantly, the function is well able to judge the relevance of information.

We may also conclude that in time-critical operations and resource constrained networks, standard communication protocols that provide transmission schedules can be enhanced by the ability to reevaluate information before every transmission attempt. In other words, it is worthwhile to combine communication protocols with RCBE.

The objective of chapter 6 was to answer the research question:

> Does the Adaptive Team Clustering help to improve the quality of the Identical Shared Awareness?

ATF determines at run-time which entities should be part of the team to create shared awareness. Results are presented for creating ISA with two ships and with three ships. From these results it is shown that ATF, depending on the benefit of coordination of actions, automatically selects the proper entities in the team that should cooperate in tracking objects. It is an addition to Request and Constraint Based Evaluation (RCBE) since the results show that there are situations when cooperation results in lower reward than when no cooperation is present, or that a subset of all acting/observing entities is cooperating.

We introduced a novel measure for evaluating the system performance in relation to its goals: the *offline reward*. We have discussed the use of α as a parameter to incorporate a measure of action effectiveness of cooperation. We argued that α works as a way for the higher level to express the action effectiveness of cooperation but that incorporating α in the utility function as one of the features of information offers the higher level a more task specifical method.

7.2 Future work

In chapter 5 we have applied RCBE to enable run-time adaptive communication on a single information abstraction level. Because multiple entities act on different levels of the information abstraction hierarchy, it is likely that they simultaneously want to communicate information. In the future we would like to investigate how RCBE can run-time determine at which information abstraction level it is most rewarding to communicate. For example, some situations in maritime operations (an example is described in the introduction) could indeed benefit from communicating tracks in contrast to single measurements. Other situations can require the transmission of single measurements.

We have utilized a rather complex communication model, where multiple parameters influence the shape of both the EDD and the cost function. We have just begun exploring the influence of these parameters on the EDD and cost function and a we would like to do a more thorough exploration in the future.

In this investigation we did experiment with information-requests that regard timeliness and accuracy of probabilistic state estimates. It would be interesting to experiment on more types of information-requests, *e.g.* requests regarding classification and to investigate how RCBE could deal with information-requests regarding *ambiguity* of information.

We have tested on maritime scenarios using radars and would like to show that RCBE can also work in other application domains like mobility and security. Another domain regards platooning in traffic, where the cars in a platoon need to maintain run-time ISA to react run-time to the dynamics within in the platoon.

Another application area where adaptive creation of ISA could be useful is in robotics. A team of robots can optimize their plans and actions given ISA. Based on the current information-requests individual robots can decide which sensory information to communicate.

The management system chooses sensors and allocates tasks for these sensors to fulfill a certain information-request. We can translate this into our terminology by saying that there is a certain information-request for a certain situation awareness.

Lastly, RCBE could be applied to *sensor management*. RCBE could calculate the reward of a sensor by balancing the value of the sensors' information and the cost of the sensors' active collection (*i.e.* energy, time). For example, when a ship is operating near the coast and needs detailed and timely positional information of a certain object on land, the ship can calculate the *expected reward* of sending out an Unmanned Air Vehicle (UAV). The UAV has the task to fly to the region where the object is, to sense the object with its camera and to send the information back to the ship. The question is therefore whether the *expected value* of the information is worth the costs (i.e. energy for flying to the region, energy for communicating the information to the ship and the delay of flying and communication).

To conclude, RCBE enables the DSS to run-time adapt communication by realistically rewarding information based on the current constraints due to communication and the current information-requests. We believe that RCBE is applicable in multiple domains, applicable for multiple information-requests and highly variable communication situations, and applicable at multiple information abstraction levels.

In future experiments we would like to test the impact on team formation of varying communication capabilities. Moreover, we would like to research the scalability of ATF by having larger groups of entities and multiple objects to track. In addition, we want to examine the adaptivity to run-time changes such as the communication capabilities or information-requests. Or the adaptivity to changing visibility, and changing effectiveness of an entity.

ATF is a framework that can be extended to other domains than tracking. It is interesting to investigates these like its application in robotics.

Another interesting extension of ATF may be the ability of hopping information via entities to other entities. Hopping may upgrade the quality of communication and hence down the communication costs and delay. It may also help in case communication is faulty.

Summary

When multiple entities, such as ships, are threatened by hostile objects they can benefit from coordinating actions towards these objects. To enable coordination the ships need to share information about these objects to create an *identical shared awareness*. Unfortunately, optimal identical shared awareness is often not possible due to limited communication capabilities, particularly when the system includes complex sensors that generate vast quantities of data. The goal of this thesis is to optimize awareness under communication constraints and four aspects were addressed to attain this goal.

The first aspect is to enable entities to estimate the current communication capabilities. A novel low-complexity communication model was developed that, firstly, could simulate different communication techniques and, secondly, could run-time estimate the *expected delay* and *expected cost* of communication. Delay is of particular interest since it has a degrading influence on the main feature of the shared awareness to be optimized: accuracy of location estimation.

The second aspect is the *expected utility* of information for the shared awareness. Entities should be able to determine the expected utility with respect to information requests. We introduced a new utility function, the *integrated utility function*, which is able to capture the subjective importance of goals. Evaluation for sharing information is based on this integrated utility function.

The third aspect is the *evaluation method* for creating the most utile awareness under communication constraints, based on the abilities of estimating the expected delay, cost and utility of information. This method evaluates the reward of sharing information with respect to the information-requests, the utility of the information and the current communication capabilities. We developed the RCBE method, which decides to share information within the team if it satisfies the information-requests sufficiently to overcome the expected communication costs.

The last aspect is *Adaptive Team Formation*, an extension of the previous evaluation method. As RBCE decides about sharing information between entities, ATF decides to include an entity in a team or exclude an entity from the team: This requires dynamic evaluation of the contribution of entities based on their current utility for the shared awareness and their current communication capabilities. Entities with a positive evaluation stayed in the team or were included in the team and entities with a negative evaluation stayed off the team or were excluded from the team.

The leading example in this thesis addressed the problem of optimizing the lo-

cation accuracy of hostile objects by multiple ships. Both the RCBE method and its ATF extension were applied in simulated scenarios. The results with RCBE showed that the method improved the shared awareness under different communication circumstances and different information-requests by selecting the most relevant information. By applying Adaptive Team Formation as well, shared awareness improved even more, because the most relevant entities for sharing awareness were dynamically selected.

In conclusion, this thesis provided a new concept for adaptively maintaining shared awareness in a multi-entity team and a method based on this concept that improved the quality of the shared awareness.

Samenvatting

Wanneer meerdere eenheden, zoals schepen, worden bedreigd door vijandige objecten is het profijtelijk om acties tegen deze objecten te coördineren. Om deze coördinatie mogelijk te maken moeten de schepen informatie over de objecten uitwisselen, zodat ze een *identiek gedeelde representatie* (*Identical Shared Awareness*) kunnen creëren. Helaas is een optimaal identiek gedeelde representatie vaak niet mogelijk vanwege de beperkte communicatiemogelijkheden, vooral wanneer de systemen complexe sensoren hebben die grote hoeveelheden data genereren. Het doel van dit proefschrift is om de gedeelde representatie te optimaliseren gegeven beperkingen in de communicatie. In dit proefschrift worden vier aspecten onderzocht om dit doel te bereiken.

Het eerste aspect is de mogelijkheid om eenheden in staat te stellen een goede inschatting te maken van de momentane communicatie mogelijkheden. Een nieuw generiek communicatie model met lage complexiteit is ontwikkeld dat zowel verschillende communicatietechnieken als de verwachte vertraging en verwachte kosten van communicatie in run-time kan simuleren. Vertraging is van groot belang omdat het een negatieve invloed heeft op het optimaliseren van het belangrijkste kenmerk van de gedeelde representatie: nauwkeurigheid van de locatie schatting.

Het tweede aspect is het *verwachte nut (utility)* van de informatie voor de gedeelde representatie. Eenheden moeten in staat zijn om het verwachte nut te bepalen met betrekking tot informatie verzoeken. We introduceerden een nieuwe nutsfunctie, de *geïntegreerde nutsfunctie*, die in staat is om het subjectieve belang van de doelstellingen mee te nemen. Evaluatie voor het delen van informatie is gebaseerd op deze geïntegreerde nutsfunctie.

Het derde aspect voor het creëren van de beste representatie gegeven de communicatie beperkingen, is de *evaluatiemethode*, die gebaseerd is op de mogelijkheden om een schatting van de verwachte vertraging, kosten en nut van informatie te maken. Deze methode evalueert de bijdrage voor het delen van informatie met betrekking tot de informatie verzoeken, het nut van de informatie en de huidige communicatiemogelijkheden. We ontwikkelden de RCBE methode, die beslist om informatie binnen het team te delen als deze in voldoende mate nut heeft voor de informatieverzoeken om de verwachte kosten voor communicatie te compenseren.

Het laatste aspect is *Adaptive Team Formation*, dat een uitbreiding is van de vorige evaluatie methode. Zoals RBCE beslist over het delen van informatie tussen de eenheden, besluit ATF een eenheid in een team op te nemen of uit te sluiten van het team: Dit vereist dynamische evaluatie van de bijdrage van eenheden op basis van hun huidige nut voor het gedeelde representatie en hun huidige communicatiemogelijkheden. Eenheden met een positieve evaluatie bleven in het team of werden opgenomen in het team en entiteiten met een negatieve evaluatie bleven buiten het team of werden uit het team gezet.

Het leidende voorbeeld in dit proefschrift betreft een probleem van het optimaliseren van de locatie nauwkeurigheid van vijandige objecten door meerdere schepen. Zowel de RCBE methode als de ATF uitbreiding werden op gesimuleerde scenarios toegepast. De resultaten met RCBE toonden aan dat de methode de gedeelde representatie verbeterde onder verschillende communicatieomstandigheden en verschillende informatie verzoeken, door de meest relevante informatie te selecteren. Door ook Adaptive Team Formation toe te passen verbeterde de gedeelde representatie verder, omdat de meest relevante entiteiten voor het delen van representatie dynamisch waren geselecteerd.

Ten slotte: in dit proefschrift wordt een nieuw concept gepresenteerd voor het adaptief bijhouden van een gedeelde representatie in een team naast een methode die op basis van dit concept de kwaliteit van de gedeelde representatie verbetert.

Bibliography

J.S. Albus and A.J. Barbera. Rcs: A cognitive architecture for intelligent multi-agent systems. *Annual Reviews in Control*, 1(29):87–99, 2005.

J. An. Empirical analyses on maritime radio propagation. In *59th Conference in Vehicular Technology*, volume 1, pages 176–180, Keelung, Taiwan, May 17–19 2011.

E. H. Aoki, A. Bagchi, P. Mandal, and Y. Boers. A theoretical look at information-driven sensor management criteria. In *14th International Conference on Information Fusion 2009*, Chicago, Illinois, USA, July 2011a.

E. H. Aoki, A. Bagchi, P. Mandal, and Y. Boers. On the 'near-universal proxy' argument for theoretical justification of information-driven sensor management. In *Proc. IEEE Workshop on Statistical Signal Processing*, Nice, France, Jun. 28–30 2011b.

Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, and Matei Zaharia. Above the clouds: A berkeley view of cloud computing. Technical report, 2009.

G. Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, March 2000.

F. Bolderheij, F.G.J. Absil, and P. van Genderen. A risk-based object-oriented approach to sensor management. In *the 8th International Conference of Information Fusion*, Seoul, Republic of Korea, 2005.

J. Boyd. A discourse on winning and losing, 1987. Maxwell AFB Lecture.

P. Brenner. A technical tutorial on the ieee 802.11 protocol, 1997. Tutorial.

Rodney A. Brooks. How to build complete creatures rather than isolated cognitive simulators. In *Architectures for Intelligence*, pages 225–239. Erlbaum, 1991.

U. Castiello and C. Umilta. Size of the attentional focus and efficiency of processing. *Acta Psychologica*, 73(3):195–209, 1990.

P. Chou and M. Zhourong. Rate-distortion optimized streaming of packetized media. *IEEE Transactions on Multimedia*, 8:390–404, 2006.

L. W. Couch. *Digital and Analog Communication Systems*. Macmillan, New York, 1993.

H.W. de Waard. *A New Approach to Distributed Data Fusion*. PhD thesis, The University of Amsterdam, 2008.

A. G. Dimakis, A. D. Sarwate, and M. Wainwright. Geographic gossip : Efficient aggregation for sensor networks. In *5th International Symposium on Information Processing in Sensor Networks (IPSN 2006)*, Nashville, TN, April 2006.

E. Dorion and A.C. Boury-Brisset. Us department of defense c4isr architecture working group: Levels of information systems interoperability (lisi). March 1998.

M. Drozdowski. Estimating execution time of distributed applications. In *Proceedings of the th International Conference on Parallel Processing and Applied Mathematics-Revised Papers*, pages 137–144, London, UK, UK, 2002. Springer-Verlag.

H. Durant-Whyte, B. Rao, and H. Hu. Towards a fully decentralized architecture for multi-sensor data fusion. In *Proceeding of the IEEE Int. Conf. on Robotics and Automation*, pages 1331–1336, Cincinnati, Ohio, USA, 1990.

M.R. Endsley. *Theoretical underpinnings of situation awareness: A critical review*. Situation awareness analysis and measurement. Lawrence Erlbaum Associates, Mahwah, NJ, 2010.

C. Eriksen and J. St James. Visual attention within and around the field of focal attention: A zoom lens model. *Perception & Psychophysics*, 40(4):225240, 1986.

S. Eswaran, D. Shur, S. Samtani, and A. Staikos. Information utility in mission-oriented networks. In *2010 IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)*, pages 1–9, Montreal, QC Canada, June 2010.

Sharanya Eswaran, David Shur, and Sunil Samtani. A metric and framework for measuring information utility in mission-oriented networks. *Pervasive and Mobile Computing*, 2011.

J. Fuster. Upper processing stages of the perception-action cycle. *Trends in Cognitive Sciences*, 8(4):143–145, 2004.

Paolo Gaudiano, Ben Shargel, Eric Bonabeau, Bruce T. Clough, and Fawcett St. Swarm intelligence: a new c2 paradigm with an application to control of swarms of uavs. In *8th ICCRTS Command and Control Research and Technology Symposium*, 2003.

B. Grocholsky. *Information-Theoretic Control of Multiple Sensor Platforms*. PhD thesis, The University of Sydney, 2002.

T. Gulrez and M. Kavakli. Sensor relevance establishment problem in shared information gathering sensor networks. In *Proceedings of the 2007 IEEE International Conference on Networking, Sensing and Control*, pages 650–655, London, United Kingdom, 2007.

D. Gunawardena, P. Key, and L. Massoulie. Network characteristics: Modelling, measurements and admission control. In *IWQos '03: Proceedings of the International Workshop on Quality of Service*, Berkeley, CA, USA, 2003.

Hamood Shehab Hamid, Dr. Widad Ismail, Ekhlas Kadhum, and Dr. J.S. Mandeep. Analyze ber performance of wireless fsk systems. *Computer-aided engineering*, November 2009. URL `http://mwrf.com/Articles/ArticleID/22151/22151.html`.

Michael Hofmann and Dariu M. Gavrila. Multi-view 3d human pose estimation combining single-frame recovery, temporal integration and model adaptation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2214 – 2221, Miami, FL, 2002.

M. Howard and R. Estkowski D. Payton. Coalitions for distributed sensor fusion. In *Proceedings of the Fifth International Conference on Information Fusion 2002*, volume 1, pages 636–642, 2002.

IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems. IEEE 802.16 Broadband Wireless Access Working Group, Apr. 8th 2002.

Multihop System Evaluation Methodology: Traffic Models. IEEE 802.16 Broadband Wireless Access Working Group, May 1^{st} 2006.

R. Jansen, W. Huiskamp, J.-J. Boomgaardt, and M. Brasseé. Real-time scheduling of hla simulator components. Simulation Interoperability Workshop, Spring 2004.

Michel C. Jeruchim, Philip Balaban, and K. Sam Shanmugan, editors. *Simulation of Communication Systems: Modeling, Methodology and Techniques*. Kluwer Academic Publishers, Norwell, MA, USA, 2nd edition, 2000.

N. Jin, G. Venkitachalam, and S. Jordan. Dynamic congestion-based pricing of bandwidth and buffer. *IEEE/ACM Trans. Netw.*, 13:1233–1246, December 2005.

R. Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASME Journal of Basic Engineering*, 82(D):3542, 1960.

T.L. Kasteren, G. Englebienne, and B.J. Krse. An activity monitoring system for elderly care using generative and discriminative models. *Personal and Ubiquitous Computing*, 14(6):489498, 2010.

F.P. Kelly, A.K. Maulloo, and D.K.H. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237252, 1998.

L.J.H.M. Kester. Designing networked adaptive interactive hybrid systems. In *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Seoul, Republic of Korea, August 2008.

L.J.H.M. Kester. *Method for Designing Networking Adaptive Interactive Hybrid Systems*, volume 281 of *Studies in Computational Intelligence*, pages 401–422. Springer, first edition, 2010.

G. Kingston and C. Martell. Degrees of shared awareness. In *Proceedings of the Ninth International Command and Control Research and Technology Conference*, Copenhagen, Denmark, September 2004.

C. Kreucher, K. Kastella, and A. O. Hero. Multi-target sensor management using alpha-divergence measures. In *Proceedings of the 2nd International Conference on Information Processing in Sensor Networks*, number 2634 in Lecture Notes in Computer Science, pages 209–222, 2003.

C. Kreucher, A.O. Hero, and K. Kastella. A comparison of task driven and information driven sensor management for target tracking. In *Proceedings IEEE 44th Conference on Decision and Control*, pages 4004–4009, Seville, Spain, Dec. 12–15 2005a.

C. Kreucher, K. Kastella, and A. Heroiii. Sensor management using an active sensing approach. *Signal Processing*, 85(3):607624, 2005b.

J.W. Lee and M. Chiang. Price-based distributed algorithms for rate-reliability tradeoff in network utility maximization. *IEEE Journal on Selected Areas in Communications*, 24(5), 2006.

S.H. Low and D.E. Lapsley. Optimization flow control,i: Basic algorithm and convergence. *IEEE/ACM Transaction on Networking*, 7:861–874, 1999.

J.W. Marck, L.J.H.M. Kester, J. Bergmans, M. van Iersel, and E. van Foeken. Effective and efficient communication of information. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2008)*, Seoul, Republic of Korea, 2008.

L. Ming, T. Yan, D. Ganesan, E. Lyons, P. Shenoy, A. Venkataramani, and M. Zink. Multi-user data sharing in radar sensor networks. In *Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 247–260, Sydney, Australia, 2007.

Arthur G. Mutambara. *Decentralized Estimation and Control for Multisensor Systems*. CRC Press Inc, 1998.

Subpanel of the Joint Directors of Laboratories. Data fusion lexicon. Tech. panel, 1991.

V. Ozdemir, S. Muthukrishnan, and I. Rhee. Scalable, low-overhead network delay estimation. Technical report, Raleigh, NC, USA, 1999.

A.K. Pradeep. *The Buying Brain: Secrets for Selling to the Subconscious Mind*. Wiley, 1st edition, 2010.

S.J. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach*, chapter 2. Prentice Hall, Upper Saddle River, New Jersey, 2 edition, 2003.

F. Sawo, F. Beutler, and U. Hanebeck. Decentralized state estimation of distributed phenomena based on covariance bounds. In *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Seoul, Republic of Korea, August 2008.

P. Scerri, R. Glinton, S. Owens, D. Scerri, and K. Sycara. Geolocation of rf emitters by many uavs. In *Proceedings of the AIAA Infotech@Aerospace 2007 Conference and Exhibit*, pages pp. 1–13, Rohnert Park, California, May 2007.

E. Setton and B. Girod. *Peer-to-Peer Video Streaming*. Springer, 2007.

Claude E. Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27:379–423, July 1948.

M.K. Simon and M. Alouini. *Digital communication over fading channels*. Wiley-Interscience, New Jersey, 2005.

Matthijs T. J. Spaan and Pedro U. Lima. A decision-theoretic approach to dynamic sensor selection in camera networks. In *Int. Conf. on Automated Planning and Scheduling*, pages 279–304, 2009.

A. Steinberg and C.L. Bowman. Rethinking the jdl data fusion levels. In *Proceedings of the Military Sensing Symposia (MSS) National Symposium on Sensor and Data Fusion (NSSDF)*, volume 2, Colombia, 2004.

THALES Land & Joint Systems. Quality of service for latency-sensitive services in a naval context, using rsvp and multicast on top of a slotted radio medium. In *RTO Information Systems Technology Panel (IST) Symposium*, Prague, Czech Republic, April 2008.

Bro@dnet information guide. Telefunken, 2008. Technical Guide.

D. Thebault. *LARA Executive Summary*, Sept. 18 2007.

E. van Foeken and L. Kester. Adaptive team formation for shared situation awareness. In *15th International Conference Information Fusion (FUSION) 2012*, pages 361 –368, May 9–12 2012.

E. van Foeken and L.J.H.M. Kester. Real-time common awareness in communication constrained sensor systems. In *12th International Conference on Information Fusion 2009*, Seattle, USA, 2009.

E. van Foeken and M.R.J.A.E. Kwakkernaat. Low-complexity wireless communication modeling for information flow control in sensor networks. In *14th International Conference on Information Fusion 2011*, Chicago, Illinois, USA, 2011.

E. van Foeken, P. Hiemstra, and L.J.H.M. Kester. An improved method for creating shared belief in communication constrained sensor networks. In *4th German Workshop Sensor Data Fusion: Trends, Solutions, Applications SDF 2009* , Lübeck, Germany, 2009.

M. van Iersel, L.J.H.M. Kester, J. Bergmans, P. Hiemstra, K. Benoist, and B. van der Broek. Creating shared situation awareness in a multi-platform sensor network. In *10th International Conference on Information Fusion 2008*, Cologne, Germany, 2008.

P. Velagapudi, O. Propopyev, K. Sycara, and P. Scerri. Maintaining shared belief in a large multiagent team. In *10th International Conference of Information Fusion 2007*, pages 1–8, Quebec City, Canada, July 2007.

H. L. Vu and T. Sakurai. Collision probability in saturated ieee 802.11 networks. In *Proc. of ATNAC, Melbourne*, pages 21–25, Melbourne, Australia, 2006.

Charles Wang, Dean Sklar, and Diana Johnson. Forward error-correction coding. *The Aerospace Corporation magazine of advances in aerospace technology (The Aerospace Corporation)*, 3(1), 2001.

H. Wang, K. Yao, G. Pottie, and D. Estrin. Entropy-based sensor selection heuristic for target localization. In *in Proceedings of the third international symposium on Information processing in sensor networks*, pages 36–45. ACM Press, 2004.

Kui Xu, Bin Dai, Fan Zhang, Benxiong Huang, and Bo Zhang. Latency estimation for time-sensitive applications under wireless network coding scheme. In *IEEE Int. conf. on Communications and Mobile Computing*, pages 263 – 266, Kunming, Yunnan, China, 2009.

S.H.K. Zhou and R. Chellappa. From sample similarity to ensemble similarity: Probabilistic distance measures in reproducing kernel hilbert space. *Transactions on Pattern Analysis and Machine Intelligence*, 28(6):917929, 2006.

List of Figures

List of Symbols

u	Action	i
t_{real}	Actual delay	i
φ	Adaptive Team Formation function	i
B	agent's bandwidth fraction	i
T	agent's timeslot fraction	i
$\mathcal{Z}_{all}$	All associated detections	i
μ_r	Average of the reference state	i
t_{wait}	back-off delay due to not receiving ACK	i
$\mathbf{J}$	Bayes expected utility	i
k	Boltzmann constant	i
$\mathcal{C}$	Costs made	i
R	data rate	i
δ	Delta dirac function	i
t_{det}	determinate/transmit latency	i
d	differential symbol	i
d	distance	i
ϕ	Domain-dependent algorithm used in the core of an agent	i
P	Effective isotropic radiated power of the transmit antenna	i
X	Environment	i
ϵ	Error of tracks	i
q	Error probability	i
Γ	evaluation parameters	i
γ	evaluation parameter	i
E	Expected	i
$\hat{c}$	Expected Cost of Communication of a single transmission attempt	i
$\hat{C}$	Expected Cost of Communication distribution	i
$\hat{P}$	Expected Cumulative Delay Distribution	i
$\hat{p}$	Expected Probability Delay Distribution	i
$\hat{R}$	Expected reward	i
$\hat{V}$	Expected value	i
α	Exponent defining the exponential increase of utility with larger teams	i
$\mathcal{F}$	Families of sets	i
t^F	frame-time duration	i
f	A function	i
$\mathcal{G}$	Gamma Cumulative Distribution Function	i
Γ	gamma distribution	i
t_G	guard time	i
l	Information abstraction level	i
Δ	Information difference	i

$\mathcal{U}$	Integrated utility function	i
t_{max}	maximum delay	i
M	Message size	i
κ	Normalization constant	i
E_b/N_0	Normalized signal-to-noise-ratio	i
g	number of detections used for include evaluation	i
Q	number of packets in a single time frame	i
F	number of time-frames needed to transmit the entire message	i
N	number of transmission attempts	i
L	packet size	i
t_p	packet transmission time	i
q_p	packet-error-probability	i
$P(a)$	power allocated for transmission to particular agent	i
$\hat{q}$	Probability of failed transmission	i
$\hat{p}$	Probability density function	i
t_{path}	propagation path delay	i
$\mathbb{R}$	Real numbers	i
$\mathbb{R}+$	Real non-negative numbers	i
G_a	Receive antenna gain	i
$\hat{x}_r$	Reference state	i
$\hat{X}_r$	Reference states	i
ψ	Request and Constraint Based Evaluation function	i
$\mathcal{R}$	Reward	i
K	Rician-K-factor	i
θ	scale parameter	i
$\mathcal{A}$	Set of agents	i
$\mathcal{Z}$	Set of detections/ the information	i
$\mathcal{J}$	Set of entities	i
$\hat{X}$	Set of identical and shared state estimates	i
$\mathcal{O}$	Set of objects	i
α	shape parameter	i
a	Single agent	i
z	Single detection	i
j	Single entity	i
$\hat{x}$	Single identical and shared state estimate	i
$\hat{y}$	Single local state estimate	i
o	Single object	i
η	Spectral efficiency	i
c	Speed of light	i
$\vec{x}$	State variable	i
$\mathcal{H}$	Subteam	i
p	success probability	i
L_{sys}	System losses	i
T_{sys}	System temperature	i

$\mathcal{S}$	Team of agents	i
n	the nth transmission attempt	i
q_F	time-frame-error-probability	i
B_{tot}	total bandwidth	i
P_{tot}	total power	i
t_{tot}^F	total frame-time	i
f	Transmission frequency	i
G	Transmit antenna gain	i
t_{TAT}	turn-around-time	i
U	utility function	i
$\mathcal{V}$	Value	i
λ	Wavelength	i

Abbreviations

ACK message of acknowledgement, page 43

ADC Analog to Digital Converter, page 38

AI Artificial Intelligence, page 16

ATF Adaptive Team Formation, page 12

BER bit-error-rate, page 39

BPSK Binary Phase Shift Keying, page 41

CS Communication Service, page 16

DCF Distributed Coordination Function, page 23

DSE Distributed State Estimation, page 21

DSS Distributed Sensor System, page 11

ECC Expected Cost of Communication, page 31

ECDD expected cumulative delay distribution, page 45

EDD Expected Delay Distribution, page 31

GCDF Gamma Cumulative Distribution Function, page 44

HLA High Level Architecture, page 31

ICP Interaction Configuration Phase, page 18

IF Intermediate Frequency, page 38

IP Internet Protocol, page 23

IRP Interaction Refinement Phase, page 19

ISA identical shared awareness, page 11

IUF Integrated Utility Function, page 73

JDL Joint Directories of Laboratories, page 18

JROADS Joint Research On Air Defense Simulation, page 31

KL Kullback Leibler, page 61

LA local awareness, page 91

LARA Layered Architecture for Real-time Applications, page 23

LOS line-of-sight, page 41

MAC Medium Access Control, page 23

MIMO Multiple-Input Multiple-Output, page 37

MSE Mean Squared Error, page 30

NAIHS Networked Adaptive Interactive Hybrid Systems, page 14

NATO North Atlantic Treaty Organisation, page 40

OA Object Assessment, page 19

OFDM Orthogonal Frequency-Division Multiplexing, page 40

OODA Observe Orient Decide Act, page 15

P2MP point-to-multipoint, page 23

pdf probability density function, page 61

POMDP Partially Observable Markov Decision Process, page 28

RCBE Request and Constraint Based Evaluation, page 12

RCI Run Time Communication Infrastructure, page 31

RF Radio Frequency, page 37

ROI Region of Interest, page 58

RTI Run Time Infrastructure, page 30

SNR Signal-to-Noise-Ratio, page 22

TAT turn-around-time, page 43

TCP Transmission Control Protocol, page 23

TNO Nederlandse Organisatie voor Toegepast Natuurwetenschappelijk Onderzoek, page 31

UAV Unmanned Air Vehicle, page 1

WiMAX World-wide interoperability for Microwave access, page 23

WLAN Wireless Local Area Network, page 23

Dankwoord

Veel mensen hebben een belangrijke rol gespeeld bij de totstandkoming van dit proefschrift. Als eerste wil ik graag mijn promotor, Frans C.A. Groen, en co-promotor, Leon Kester, bedanken. Frans, zonder jou had ik het niet gered. Je was een goede, volhardende coach die mij tijdens elke meeting scherpe inzichten leverden. Toen mijn aanstelling afliep en ik ging werken als Software Engineer, hielden wij met regelmaat contact. Zo nu en dan kreeg ik een (voorzichtige) mail met de boodschap: maak het af! En zelfs een diner met co-promoter Leon werd ingezet als wapen voor de afronding. Uiteindelijk waren je onmetelijke geduld, positivisme en vertrouwen de juiste ingrediënten om het na al die jaren af te ronden. Mijn eeuwige dank daar voor!

En dan Leon, ik omschrijf jou graag als mijn inspiratiebron. Ik ken weinig mensen die met zo'n grote passie over een onderwerp kunnen praten, en dit werkte aanstekelijk op mij. En ten tweede ben je de uitvinder van het NAIHS Model, waarop mijn onderzoek is gestoeld. Je zou dus kunnen zeggen dat zonder jou model mijn proefschrift niet zou kunnen bestaan. En ten derde heb jij mij kennis laten maken met de filosofen Spinoza en Kurzweil. Spinoza is dan wel niet direct relevant voor mijn onderzoek, maar het is wel een verrijking voor mijn geest geweest.

Natuurlijk wil ik ook Maurice Kwakkernaat graag bedanken voor het grote aandeel in hoofdstuk 3 van het proefschrift. Samen hebben wij het artikel geschreven dat aan de basis ligt van dit hoofdstuk. Je was in die tijd altijd beschikbaar voor feedback, en het was fijn om samen met je te werken.

Ook wil graag Jacob van der Pol bedanken. Een groot deel van de code die het mogelijk maakte voor mij om simulaties te runnen, is geschreven met jouw pen.

De leescommissie, bestaande uit professor de Laat, professor Adriaans, professor Fledderus, professor Van den Berg en doctor Visser wil ik graag bedanken voor de tijd en moeite die ze hebben genomen om mijn proefschrift te lezen en te beoordelen.

Jan Willen Marck, oud studiegenoot, ook jij kan niet gemist worden in het dankwoord. Jij hebt mij laten kennismaken met Ultimate Frisbee, mijn grote passie. Maar dat niet alleen, jij hebt mij ook getipt dat ze op de afdeling 'Distributed sensor systems' van TNO een promovendus zochten voor dit onderwerp, wat ik met beide handen heb aangegrepen. Je zou dus kunnen zeggen dat jij bepalend bent geweest voor een paar belangrijke keuzes mijn leven.

Mijn collega's op TNO in Den-Haag en dan vooral aan afdelingsgenoten Peter

Hiemstra, Jan Willem Marck, Jeroen Bergmans, Abdelmahjid Salah, Johan van der Pol heb ik goede herinneringen. Jullie brachten ontspanning tijdens de pauzes en het kaartspel Dalmuti werd een uur fanatiek gespeeld. Soms leidend tot lichte irritatie van het hoofd van de afdeling, omdat wij toch weer een keer aan het werk zouden moeten gaan.

Natuurlijk ben ik mijn goede vrienden, tevens oud studiegenoten, Jeroen van Dijk en Joris IJsselmuiden, zeer erkentelijk dat ze mij hebben bijgestaan tijdens het uur van de waarheid.

Ook kan ik zeggen dat ik het proefschrift een beetje wil opdragen aan goede vriend Jonathan Stoltzfus. Elke keer dat we elkaar zagen zei je zoiets als *You are going to finish right, I will be invited no, I have never been at a promotional event, O man I would love to come*.

En dan wil ik graag mijn allerlaatste woord richten tot mijn grote liefde: Nynke. Je hebt een essentiële rol gespeeld in de voorbereiding van de verdediging. Je hebt me kort gehouden. Streng geweest in dat ik niet naar Ultimate Frisbee toernooien mocht gaan. Zorg gedragen voor kleine Abe. En bovenal, heb je me regelmatig ondervraagd over het onderwerp zodat ik makkelijker kon spreken over de theorie. Ik hoop dat je me nog veel wilt helpen in de toekomst, want ik weet zeker dat ik die goed kan gebruiken.