Maximising Effectiveness of Distributed Mobile Observation Systems in Dynamic Situations

Leon Kester, Maarten Ditzel
TNO, Oude Waalsdorperweg 63, 2597 AK The Hague, The Netherlands
Email: leon.kester@tno.nl

***Abstract*—The trend in modern day observation systems is towards distributed (often mobile) systems that are able to automatically adapt themselves in dynamic situations. They have to make most of their resources to maximise the system's effectiveness, all at reasonable cost. Currently, there is no formal framework of reasoning to design and implement the mechanisms for the required adaptivity. In this paper, utility based reasoning is combined with the Markov Decision making Process (MDP) formalism to come to such an approach. Its ramifications are discussed, followed by several cases in which this approach has been applied successfully.**

***Index Terms*—fusion process management; resource management; distributed fusion; system modelling; autonomous systems**

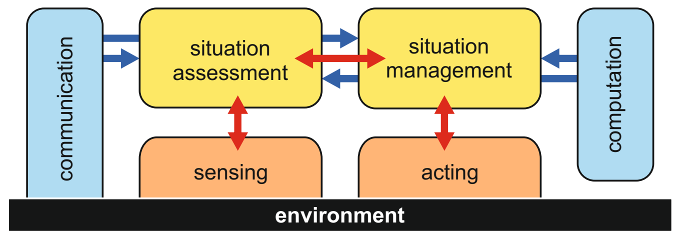

Figure 1. High level Cognitive system model.

I. Introduction

Modern day observation systems are expected to run reliably in constantly changing conditions and increasingly complex environments to provide their users with actionable information. These cognitive systems are required to automatically adapt themselves to (i) changes in the observed situation, (ii) changes in the system's environment, (iii) changes in the information needs of the user, and finally (iv) also changes within the system itself. If the system has an abundance of resources (for example energy, or computation power) at its disposal, this is already a complex task. However, when resources become scarce, it becomes a daunting undertaking.

The above is especially true for distributed mobile systems, where one has to take into account the (usually limited and volatile) communication means between the different parts of a system. If these systems are to effectively and efficiently act on the aforementioned changes, they have to implement some level of self-awareness and self-management. Unfortunately, there is no systematic approach for this.

For self-management, different techniques are available. Some of these are based on using low level information theoretic measures [1]–[3]. However, utility based measures are more appropriate, as they are quantitatively related to the effectiveness to the system's goal. Such measures are applied in [4], [5] for sensor management and in [6] for the classification of data flows. In [7]–[9], the authors demonstrate the advantage of using utility-based metrics for optimizing communication constrained systems.

In [10], we have explored the idea of generating utility-based metrics at run-time to optimize a multilevel fusion system. The paper addresses a scenario where a high value unit needed protection, utilizing a computation and communication constrained distributed observation system, including system management tasks such as action planning. We also have successfully used a similar approach for applications in the mobility domain. The switch of domains enticed us to further develop our work on run-time utility based system management. In this process, we realised that the underlying mechanism is generally applicable to any function in the system. Moreover, it provides us with a selection mechanism to choose from a large set of possible actions to take when managing the observation system.

Consequently, in this paper we would like to make a step forward in formalising the run-time utility based system function management method. For this we adopted the formalism as used by a Markov Decision making Process (MDP). It provides us with a reasoning framework for the observation and management of both the external situation, as well as the internal situation. It provides a solid foundation to implement the system's desired level of self-awareness and self-management in a systematic manner.

In Section II we first introduce a high level model for cognitive systems that forms the basis for the subsequent introduction of a management function (Section III) to optimize the system's effectiveness. Several refinements to the approach are discussed in Section IV to come to a more generally applicable reasoning framework for mobile distributed observation systems. Finally, the framework's use is demonstrated in several use cases. In Section VI we conclude this paper.

II. Cognitive System modelling

In many different research areas various ways to model systems with cognitive capabilities have been proposed e.g. [8], [11]–[15]. From these, we derive a high level cognitive model as depicted in Fig. 1. In this model we identify the primary functions Situation Assessment (SA) and Situation

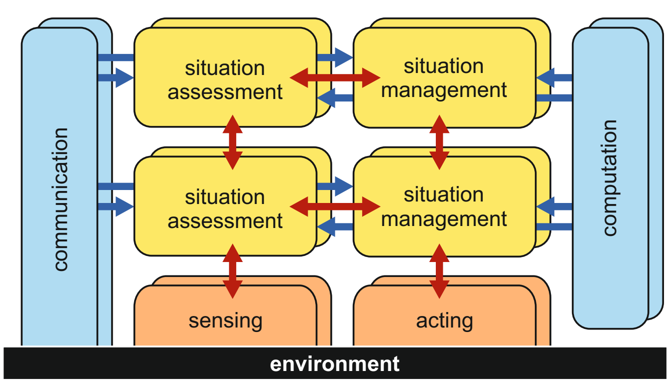

Figure 2. Distributed Hierarchical Cognitive system model.

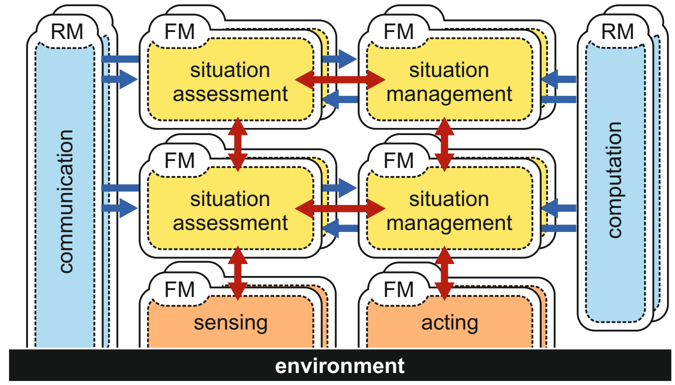

Figure 3. Adaptive Distributed Hierarchical Cognitive system model.

Management (SM)[1], Sensing, Acting and the supporting Communication and Computation services. The well-known OODA loop is easily recognized following the sequence of primary functions: Sensing, SA, SM and Acting. The system interacts with the environment through its sensors and actuators. Moreover, the communication function may also influence or be influenced by the environment in which the cognitive system operates.

For many applications the functions of the cognitive system are distributed over many physical entities with considerable communication constraints. Moreover, in many of the aforementioned cognitive models a hierarchy of SA and/or SM functions is applied. The nature of such a hierarchy may depend on the specific application, but usually a hierarchy addresses one or more of (i) different levels of abstraction in representing the situation, (ii) different time scales for which situation management is considered, and (iii) different geometric scales over which management is coordinated. For the cognitive system model used in this paper this means that there are multiple instantiations of the same type of functions. The resulting distributed hierarchical system model is depicted in Fig. 2.

Finally, many cognitive system models also foresee some form of management of the internal functions. Unfortunately, there is little consensus on which functions should be managed, what elements of those functions should be managed, and if the function management should be implemented in a distributed way, i.e., if the system is managed in a centralised way, or that each function has its own manager.

In the cognitive system model we take as starting point a fully distributed implementation, where each function has its own local management. We consider a more centralised approach only if certain management actions have consequences for multiple functions, which cannot be effectively addressed independently. The local function manager (FM) is responsible for maximising the effectiveness of the managed function. The function managers that manage the communication and computation resources we call resource managers (RM). The resulting adaptive model is depicted in Fig. 3. The role of the function manager is discussed in detail in the next section.

[1]The term *situation* as used in SA and SM refers to the cognitive system together with those parts of its environment that are relevant to the system's users and the operation of the system itself.

III. Function Management

As stated, the function manager has to maximise the managed function's effectiveness. Thereto, it has to decide what actions are likely to contribute most, making the best out of the available time and resources. For this, it of course needs expressions for deciding which action is better than others.

To make the problem more tangible, let us now consider management of one specific SA function. Function management can consider many different types of actions to perform on this SA function. A non-exhaustive list of actions to consider are:

- Select which information to process.
- Adjust algorithmic parameters.
- Adjust model parameters.
- Communicate information to another function.
- Select different algorithms to implement the function.
- Select different models for representing the situation.
- Reorganize internal sub-functions.
- Change interaction between internal sub-functions.

As was mentioned in the introduction we adopt a Markov Decision making (MDP) framework to reason about the most rewarding actions. For now we assume that (i) the behaviour of the SA function is known, that (ii) the actions being considered are discrete, that (iii) there is no uncertainty in the situation observable by the cognitive system and, finally that (iv) the reward of the actions is immediate.

These are strong assumptions that are generally not valid in practical situations. However, for clarity we have chosen to introduce reasoning about the rewards of the various actions first. Once this approach is clear, we then discuss how the restrictive assumptions can be alleviated through several refinements. These refinements will be discussed in Section IV.

A. Reward of actions

Central to our approach is the calculation of the reward of possible actions to take. Therefore, we start with its mathematical derivation, integrating the notion of the utility of information with the MDP formalism.

Following the MDP framework, an action a will change the internal state of the function, s to s' and is, assuming for the moment a deterministic process, given by the transition function T:

$$s' = T(s, a). \tag{1}$$

The new output, x' is then given by:

$$x' = X(s'), \tag{2}$$

where X is a function that derives x from s. With this we can calculate the utility U_a (also called benefit, effectiveness or value) of action a, when the function is in state s and our current belief is x, i.e.,

$$U_a(s, a) = U_i(X(T(s, a))|x). \tag{3}$$

Here U_i is the utility function for new information, that is provided by the function that uses the new output x'. It is described in more detail in Section III-B.

Apart from the action's utility, we also need to estimate the action's costs. In this paper, we focus on the cost of computation (processing) and communication, but other costs like energy consumption, memory use or sensing time can easily be included in the same manner.

The computation cost, C_a^p, involved in doing action a is given by:

$$C_a^p = C_p(K(s, a), d_p), \tag{4}$$

where d_p is the delay due to the processing and K the amount of processing needed when doing action a in function state s. Finally, C_p is the cost function provided by the computation function.

Next to processing, additional communication costs have to be taken into account for sending x' to any consumers. This cost C_a^c is given by:

$$C_a^c = C_c(B(x'), A_n, d_c) \tag{5}$$

where d_c is the communication delay, B the bandwidth needed for transmitting x' and A_n the configuration of SA and SM functions to whom x' has to be sent to. Again, C_c is the cost function provided by the communication function. Using an MDP formalism we can now calculate the reward R of the action:

$$\begin{aligned} R(s, a) =& U_a(s, a) - C_a^p - C_a^c \\ =& U_a(s, a) - C_p(K(s, a), d_p) - C_c(B(x'), A_n, d_c). \end{aligned} \tag{6}$$

Due to the delays in communication and computation, reception of x' will be delayed by the sum of these delays. This total delay d has to be taken into account when calculating U_a. Consequently, the expression of reward R becomes:

$$R(s, a, d) = U_a(s, a, d) - C_c(K(s, a), d_p) - C_c(B(x'), A_n, d_c) \tag{7}$$

Now, it is possible to systematically value and consistently compare actions based on their rewards.

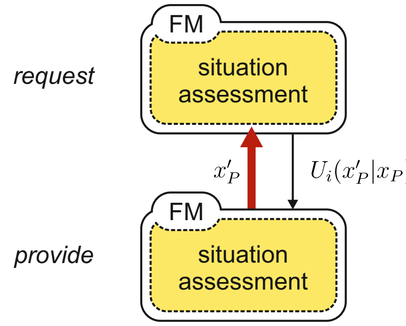

Figure 4. The requesting function provides the information provider with the information's utility U_i.

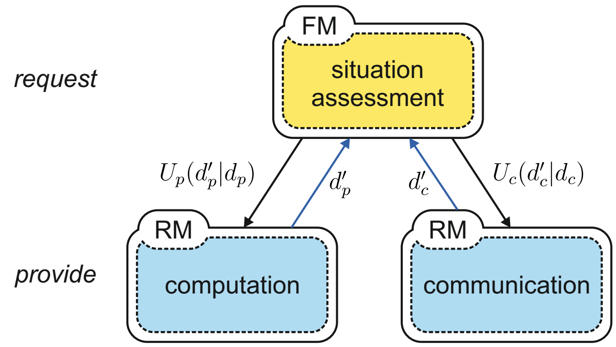

Figure 5. The requesting function provides the resource provider with the resource's utility (U_p or U_c).

B. Management of information

Key element in the calculation of the reward of a certain action (Eqn. (7)) is the utility of an action U_a that depends on the utility of new information U_i. For each flow of information between functions, the FM of the requesting function has to provide a corresponding utility function to the providing function (see Fig. 4). Given the current value of an input x_P, the FM can calculate U_i for every possible value of x'_P, using:

$$U_i(x'_P|x_P) = R(s, a_P, d) - R(s, a_0, d). \tag{8}$$

Here, a_P is the action that corresponds to updating the requesting function's state s by processing the provided x'_P. Similarly, a_0 corresponds to updating s without processing x'_P, i.e., using the old x_P. It should be noted that the requesting and providing information have their (independent) needs for communication and computation resources. Therefore, in Eqn. (8) the utility is based on the difference in reward and not only on the difference in the utility of the action.

C. Management of resources

The expressions in the previous chapter also facilitate formalising the management of the resources. Using the expressions, it is possible to calculate the effect on reward R when the communication or computation service improves, e.g., when the processing or communication delay is reduced. As depicted in Fig. 5 and similar to the use of a utility function, the requesting function managers may generate a utility function for processing U_p or communication U_c:

$$U_p(d'_p|d_p) = R(s, a, d'_p) - R(s, a, d_p), \tag{9}$$

and

$$U_c(d'_c|d_c) = R(s, a, d'_c) - R(s, a, d_c), \tag{10}$$

Reducing the delay may come at an extra cost. Thus, also the resources themselves could be equipped with a *resource manager* (RM) that reasons what actions to take to maximize the resource's reward.

D. Management of Sensors and Actuators

Sensors and actuators can be considered as low level SA and SM functions. Therefore, they should be managed in a similar way as the SA and SM functions. The main difference is that the cost, just as in the case of managing communication and computation resources, is primarily dependent on the energy and time that are needed to take the actions.

E. Energy management

Just as system function management can be extended to communication and computation management, these can be subsequently extended to power management. The power manager can reason about the reward of making extra costs to generate more power. This approach for power management is very similar to power matching methods used in smart grids. In this case, however, the goal is to maximize reward, which is the ultimately cost-effectiveness of the cognitive system as a whole, reflected by a utility function for power.

F. Planning

The calculations presented in the previous chapters should be considered over a certain period of time. This can be very short, more or less instantaneous, as is the case of an action that decides on the processing of new information. Other actions such as selecting another algorithm or models will have to be considered over a longer period of time. Another possibility is that a whole sequence of actions in a specific order (i.e., a plan) needs to be considered[2]. The calculations on reward can be gracefully extended to planning, following the MDP framework by considering the reward of a sequence of actions. This is also known as the *return* of the plan.

G. Difference between function management of SA and SM

Until now we have focussed on the management of SA functions. At this point we like to extend the discussion to include SM as well.

The output of the SA functions are best estimates of the past and current situations, while the output of the SM functions are the intended actions and best estimates of the future situations taking the intended actions into account. This information goes from SA to SM in multiple levels of hierarchy as depicted in Fig. 6. The SA functions may require input from the lower level SA_- or from the same level SA_o while the SM functions require information from SA at the

[2]Systems that do not consider this sufficiently are also called myopic.

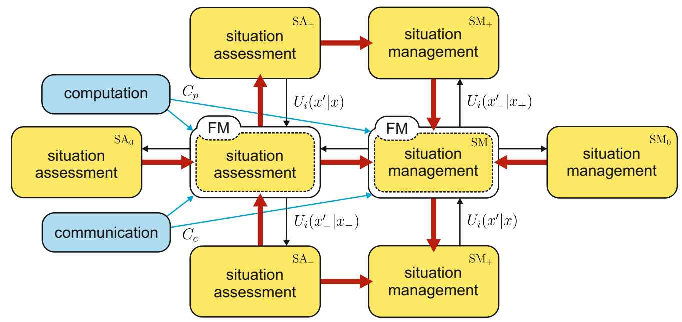

Figure 6. Adaptive resource and information management.

same level. Moreover, they may require intended action plans and expectations about future situations from the same level SM_o or a higher level SM_+. This difference is reflected in the directions of information and utility functions between SA and SM depicted in the figure. It should be noted that, although the directions are different, this does not lead to a different way in which the rewards of self-managing actions are calculated.

IV. Refinements

With the framework of reasoning described in the previous chapter it is in principle possible to reconfigure and tune the cognitive system during run-time, such that effectiveness is maximized under all circumstances. However, in practice quite some complications arise such that we can only approximate maximum cost-effectiveness. In this chapter we discuss a few of these complications and how these can be resolved through refinements of the reasoning framework.

A. Common currency for utility and cost

In order to calculate the reward of an action, the quantitative unit of utility (its currency) and ultimately the currency of effectiveness has to be the same as the currency of cost. In practice this is quite often difficult to establish. An alternative approach is that we define the reward to be the actions' utilities per unit of cost and then select the highest rewards that can be accommodated by the resource. In this approach we either have to maximise utility with fixed resources or minimise cost with a fixed goal.

We should realize that the aforementioned approach can only be adopted if the resources have a certain fixed potential, or if the utility function is a kind of step function such that the effectiveness of the system is either good or not good (utility zero). Instead, if the circumstances are such that we can improve the system's effectiveness with a run-time improvement of the resource's quality of service at certain costs, then we somehow have to express both cost and utility in the same currency.

B. Uncertainty in x

In the formulations presented in Section III uncertainty was not considered. However, in most practical cases there will be uncertainty in x and x'. Because of these uncertainties we will have expected utilities, expected costs and expected rewards of actions.

In this case, the expected utility $\hat{U}_i(x'|x)$ of new information x' given prior information x can be calculated using

$$\hat{U}_i(x'|x) = \int_{X'} \int_X U_i(x'|x) P(x'|x) P(x) dx' dx, \qquad (11)$$

where $P(x'|x)$ and $P(x)$ denote the probability distributions of x' (conditional on x) and x, respectively. X' and X contain the range of all possible values for x' and x. It should be noted that both x' and x possibly represent multi-dimensional information elements.

In addition, it should be noted that the expression for the expected utility resembles the expression for the Kullback-Leibler divergence (or distance) which is often used as a measure of performance. However, instead of using an information theoretic measure, here utility of information is used to weigh the divergence in x.

If the same approach is use as described in Section III $\hat{U}_i(x'|x)$ should be calculated for all possible values of x' and sent to the function that provides x'. This is for all practical purposes not very efficient.

Two possible solutions are suggested. The first is to send $U_i(x'|x)$ instead of $\hat{U}_i(x'|x)$. The value of Eqn. (11) with specific outcomes of x' on specific actions is then calculated in the function providing x'. The second solution is to calculate the derivatives of $\hat{U}_i$ with respect to x and with respect to an uncertainty parameter (e.g., the standard deviation σ_x for Gaussian distributions) and send these values. If this last method is adopted, precautions should be made to ensure that the utility function is well behaving in the range of possible values of x' and σ'_x.

C. Uncertainty in communication (delay and packet loss)

In most communication systems there is an uncertainty in the expected delay of a transmission, d_c. Therefore, if the action's utility U_a significantly depends on the actual delay, both the action's utility and cost need to be integrated over the expected delay distribution during the calculation of the action's reward. In this way also packet loss can be taken into account since this can be reflected in the distribution as a reduction of the probability of successful communication $P_c(d_c)$.

The expected reward of communicating x' is then given by

$$\hat{R}(s,a) = \sum_{d_c} P_c(d_c)(U_a(s,a,d_c) - C_c(B(x'), A_n, d_c)). \qquad (12)$$

If the (in this case discrete) delay distribution $P_c(d_c)$ is due to several consecutive attempts to transmit x', after each attempt one can re-evaluate the reward to determine if new attempts are still beneficial. Then, there will be a finite probability that communication costs will be made without a contribution to the action's utility. In this case Eqn. (12) is slightly modified

$$\hat{R}(s,a) = \sum_{d_c}^{d_c^{max}} (P_c(d_c)U_a(s,a,d_c) - C_c(B(x'), A_n, d_c)) \qquad (13)$$

to accommodate for the finite number of communication attempts.

D. Continuous action space

If the action space is continuous or very large simply calculating the reward in not feasible and some kind of approximation or optimisation method needs to be applied. It may, however, turn out to be useful to select the optimisation method and when to apply it, on the basis of the highest expected reward.

E. Uncertainty in state transition

In some cases there is uncertainty about how an action will change the internal state of the function. In that case a stochastic formulation of the state transition is used where T includes the probability of transition from s to s' given a certain action.

F. Uncertainty in s

In Section III-A, it was implicitly assumed that the FM has full knowledge of the function's current internal state s. However, sometimes it can be difficult to know the current state s of the function. This may for instance occur when the function is implemented in dedicated hardware accelerators and the FM does not have access to its internals. In this case, additional functionality has to be implemented as part of the FM with the sole purpose to estimate the function's internal state.

Still, with the added functionality, the state of the function s may remain not fully observable. Therefore an additional uncertainty in s' and therefore also x' is introduced. In MDP terminology, the state is only partially observable and we could use a partially observable MDP (POMDP) formalism. In this case, one might also want to adopt learning methods to reduce the uncertainty in s.

G. Efficiency of sending requests

The presented framework of reasoning readily fits a service oriented architecture approach. The service consumers generate a (utility) request function that is sent to the providers. The providers try to maximize their service. The underlying assumption is that the providers have maximum autonomy and the communication costs of the induced overhead stay low. In some cases this assumption does not hold and the communication of the utility function lead to a substantial communication load between service consumer and provider. Then, the type of interaction needs to be tuned to reduce the cost of communication, for instance by making use of predefined templates for representing the utility functions.

H. Dealing with asynchronous input and/or varying uncertainty in x by SM functions

Some type of SM functions, in particular those applying closed loop control algorithms, cannot handle asynchronously arriving information properly. The same is true for various control loops, whose inputs have a varying uncertainty. In this case the utility request function needs to take the stability of the control loop into account.

V. Applications

The utility based approach has already been applied for several applications. These have been described in various publications. However, the approach and terminology across the papers is not fully consistent with what has been described in this paper. Therefore, we want to briefly discuss a number of these applications, but now in a formulation consistent with Section III. In addition we discuss new developments for those applications and research in new application domains. Our purpose is to show that for a wide variety of applications the presented approach is applicable and useful. Detailed results for this research are and will be published in separate papers.

A. Shared awareness in a maritime environment

In [7] a utility based method is applied, called request and constraint based evaluation (RCBE) for establishing a common awareness on a task group of frigates. Only a single level in SA is considered (non hierarchical): kinematic state estimation or tracking of the relevant objects (fighters) in the environment. The requirement is to have an identical kinematic picture at all platforms/frigates.

Limited communication is the dominating factor and therefore the single cost taken into account. The action considered by the function manager on each platform is whether to send a new measurement to all platforms or not. Uncertainty in x is taken into account according to Eqn. (11). The uncertainty is calculated at the tracking level based on a received utility function $U_i(x'|x)$. In this case, however, each platform has more information available than what is being exchanged. Therefore the local awareness of the kinematic state of an object is usually (marginally) better than the common shared awareness that can be established. Therefore, a slightly different formulation was used for the action's expected utility, i.e., to communicate a new measurement:

$$\begin{aligned}\hat{U}(s,a) =& \hat{U}_i(x'|x) \\ =& \int_{X'}\int_{X_r} U_i(x'|x_r)P(x'|x_r)P(x_r)dx'dx_r - \\ & \int_{X'}\int_{X_r} U_i(x|x_r)P(x|x_r)P(x_r)dxdx_r.\end{aligned} \tag{14}$$

Here, x_r is the reference state, the local best estimate of the kinematic state of the object. In other words we calculate the difference in utility of the old state x and the new state x' with respect to x_r.

Uncertainty in the communication delay was taken into account in accordance with Eqn. (13). We model the communication system using an expected delay distribution. Integration is done over the delay uncertainties. Additionally, after each failure in trying to communicate, the reward is re-evaluated to determine if it is still rewarding to keep on trying.

In the harsh maritime environment it is assumed that for optimal shared awareness a multi hypothesis tracking (MHT) method is required. Therefore, in current research we adapt the RCBE method to work with this type of tracking algorithm. In addition we are developing utility based multi-hypothesis multi-platform track initiation methods.

B. Adaptive team formation

In [9] we employ the aforementioned RCBE method. In addition, we consider the action of including or excluding a platform from the RCBE group in tracking objects in the environment. This evaluation is done for each object track separately. Each platform calculates for each track two rewards. The first reward for tracking the object with the RCBE group jointly. The second for tracking the object only locally, while only the RCBE track result of the rest of the RCBE group is received. In order to avoid platforms joining and leaving the RCBE object tracking group on a very short time basis, the reward is integrated over a sizeable time scale with a set of Z_n measurements in this time window.

The two rewards are:

$$\hat{R}_{inc}(\hat{x}_0, Z_n, A_n) = \sum_{z_n \in Z_n} R(\hat{x}_n, z_n, A_n) \tag{15}$$

and

$$\hat{R}_{exc}(\hat{x}_0, Z_{n-1}, A_{n-1}) = \sum_{z_{n-1} \in Z_{n-1}} R(\hat{x}_{n-1}, z_{n-1}, A_{n-1}) \tag{16}$$

added with

$$\hat{R}_{exc}(\hat{x}_0, Z_1, A_1) = \sum_{z_1 \in Z_1} R(\hat{x}_1, z_1, A_1) \tag{17}$$

The decision if the platform should be part of the RCBE group to track the object depends on if

$$\begin{aligned}\Delta\hat{R}(\hat{x}_0, A_{n-1}, A_n) = \hat{R}_{inc}(\hat{x}_0, Z_n, A_n) - \\ \hat{R}_{exc}(\hat{x}_0, Z_{n-1}, A_{n-1}) - \hat{R}_{exc}(\hat{x}_0, Z_1, A_1).\end{aligned} \tag{18}$$

is positive or negative. If the value is positive the platform should be part of the RCBE group. If it is negative it should not. In this case also utility is attributed to all platforms having identical awareness. Therefore, in case the platform is not part of the RCBE group, an extra utility penalty must be added for the difference in awareness of the object between the RCBE group ($\hat{x}_{n-1}$) and the local platform ($\hat{x}_1$).

C. Autonomous vehicle mission planning

In [16] we applied a utility based approach for a search and identify mission of an autonomous Unmanned Areal Vehicle (UAV). The goal of the UAV is to detect as many objects as possible in a heterogeneous terrain with a variable probability of detection. Simultaneously, the UAV aims to identify detected objects more accurately.

The cost in this case is fuel consumption, resulting in a finite flight path length. In addition, there is an information utility function $U_i(x, t_f)$ representing the utility of detecting an object in state x that may depend also on the terrain features t_f and there is a utility function $U_i(x'|x)$ that states the utility of increasing the knowledge of the detected object from x to x'.

Before the UAV is deployed, a particle swarm optimizer algorithm and a priori knowledge of the terrain is used to find the most utile path for the projected path length. When the

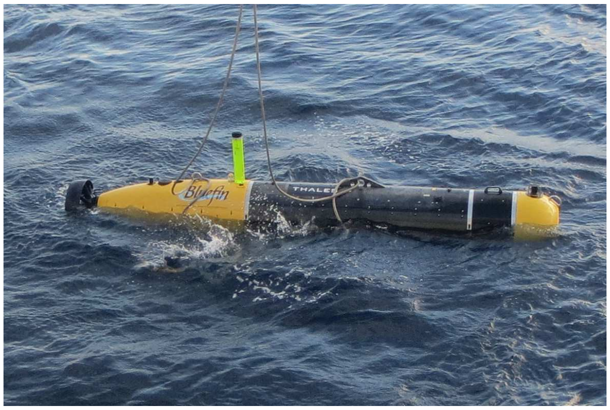

Figure 7. The MUSCLE AUV used during the sea trials.

UAV discovers an object it compares the expected rewards of continuing the predefined path or flying an extra round over the just detected object in order to increase the knowledge of the object detected. In mathematical terms

$$\Delta\hat{R} = U_i(x'|x) - \Delta L_p P_d(x, r_f) U_i(x, r_f) \qquad (19)$$

with ΔL_p the decrease in detection flight path due to the extra round over the just detected object and P_d the detection probability of objects in that flight path area. If $\Delta\hat{R}$ is positive the extra round in made, otherwise the flight is continued as planned.

Building further on this research we have applied and implemented utility based methods for optimising naval mine hunting with autonomous underwater vehicles (AUV). During sea trials, an AUV (see Fig. 7) successfully optimised its survey pattern (see Fig. 8) to improve the probability of correct identification, with consumed battery power as its main cost.

Currently we are investigating how to combine mission planning and situation assessment capabilities on one platform and how a swarm of autonomous vehicles can coordinate their behaviour such that the effectiveness of the swarm as a whole is maximised. Here, we also take the cost of communication into account.

D. Cross layer optimization in hierarchical chains

In [10] the main objective is to show the benefits of feedback with utility functions in an hierarchical chain of cognitive functions. The use case is the protection of a high value asset that is threatened by UAVs. Uncertainty of x is taken into account using Eqn. (11). For efficiency reasons, the derivative of the utility functions with respect to x and σ_x are sent, as explained in Section IV-B. Note the different notation in [10]. Value (V) is used for utility and utility (U) is used for the derivatives. The cost of communication and computation is not taken into account for each individual SA level, but only for the hierarchical SA/SM chain as a whole.

E. Utility based information management for mobility

Development in technology will result in more intelligent and autonomous vehicles being managed and supported by

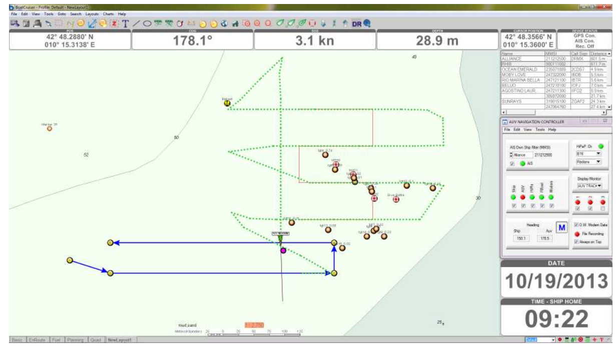

Figure 8. Screen dump of the AUV monitoring application during the execution of an autonomously planned survey.

increasingly intelligent road side units. This will lead to a higher demand for communication of information for various applications in this domain. Since the utility of information will depend on the particular traffic situation a utility based approach seems appropriate.

A use case we are focussing on is cost-effective communication for cooperate adaptive cruise control (CACC) in particular for a 'safety-checking' collision avoidance algorithm [17] we developed. Other use cases are merging of an unequipped vehicle into a string of vehicles equipped with CACC and ramp metering with unequipped vehicles. For these applications intelligent vehicles and road side units determine, depending on the available information and the utility of that information, what is most rewarding to communicate.

F. Smart grids

The objective in smart grids is to balance the energy use of all producers and consumers at multiple time- and geographic scales. Therefore, management systems usually are organised according to Fig. 2. The SM functions operate on different energy markets an therefore need to plan their actions. The larger the market that needs to be balanced, the higher the expected cost of communication in order to coordinate the behaviour of the different SM functions. For this reason a utility based information management method was introduced that not only minimises on energy cost but on the sum of energy cost and communication cost.

G. Other applications

In addition a utility based approach is considered for other applications such as multi-camera surveillance, greenhouse temperature monitoring and control and naval ship automation. Also in these cases it is foreseen that balancing cost of sensing, communication and/or computation (or underlying cost of energy) with utility of resulting cognitive capabilities during operation will result in more cost-effective systems.

VI. Conclusions

In this paper we presented a formal framework that can be used to design, develop and implement systems that can manage the cognitive functions and the resources needed for those functions at run-time. The framework combines utility based reasoning with the Markov Decision making Process (MDP) formalism. The framework enables local cognitive function managers to pursue local goals in the form of utility functions. However, since utility reflects the increase in effectiveness of the cognitive systems as a whole, pursuing these local goals leads to a more effective system.

It is fully acknowledged that, although the framework seems straight forward to apply, many complications can arise such that only approximate methods can be used. A number of such complications and the subsequent refinements of the model are discussed in Section IV. However, we believe that better systems will result when these complications are consciously dealt with, instead of being discarded as too complicated with unknown consequences. In Section V a number of cases is discussed where the approach has been successfully applied.

Due to the successful application of the proposed approach in different domains we are confident that it is a valuable extension in the design, development and implementation of distributed adaptive cognitive systems.

ACKNOWLEDGEMENT

Part of this research was conducted in the project Sensor Technology Applied in Reconfigurable Systems (STARS) funded by the Dutch Government. The authors would like to thank Stefania Giodini and Bardo Bakker for their valuable contributions.

REFERENCES

[1] C. M. Kreucher, K. Kastella, and A. O. H. III, "Sensor management using an active sensing approach," *Signal Processing*, vol. 85, no. 3, pp. 607–624, 2005.

[2] J. L. Williams, "Information theoretic sensor management," Ph.D. dissertation, Massachusetts Institute of Technology, 2007.

[3] P. A. Ortega and D. A. Braun, "A conversion between utility and information," in *3rd Conference on Artificial General Intelligence*, March 2010.

[4] F. Bolderheij, "Mission-driven sensor management, analysis, design, implementation and simulation," Ph.D. dissertation, Delft University of Technology, 2007.

[5] A. Charlish, "Autonomous agents for multi-function radar resource management," Ph.D. dissertation, University College London, 2011.

[6] S. Eswaran, D. Shur, and S. Samtani, "A metric and framework for measuring information utility in mission-oriented networks," *Pervasive and Mobile Computing*, vol. 7, no. 4, pp. 416–433, August 2011.

[7] E. van Foeken, L. Kester, and M. van Iersel, "Real-time common awareness in communication constrained sensor systems," in *12th International Conference on Information Fusion*, July 2009, pp. 118 – 125.

[8] L. Kester, "Method for designing networking adaptive interactive hybrid systems," in *Interactive Collaborative Information Systems*, R. Babuska and F. Groen, Eds. Heidelberg: Springer-Verlag Berlin, 2010, vol. SCI 281, pp. 401 – 421.

[9] E. van Foeken and L. Kester, "Adaptive team formation for shared situation awareness," in *15th International Conference on Information Fusion*, July 2012, pp. 361–368.

[10] M. Ditzel, L. Kester, S. van den Broek, and M. van Rijn, "Cross-layer utility-based system optimization," in *16th International Conference on Information Fusion*, 2013, pp. 118 – 125.

[11] J. Boyd, *A Discourse on Winning and Losing*, 1987.

[12] J. S. Albus, "Outline for a theory of intelligence," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, no. 3, 1991.

[13] R. Brooks, *How to build complete creatures rather than isolated cognitive simulators*, K. VanLehn, Ed. Hillsdale, NJ: Lawrence Erlbaum Associates, 1991.

[14] J. M. Fuster, *Cortex and Mind: Unifying Cognition*. Oxford University Press, 2003.

[15] A. Steinberg and C. Bowman, "Rethinking the JDL data fusion levels," in *Proceedings of the national symposium on Sensor Data Fusion JHUAPL*, June 2004.

[16] W. van Willigen, M. Schut, A. Eiben, and L. Kester, "Online adaptation of path formation in search and identify mission," in *10th International Conference on Adaptive and Natural Computing Algorithms*, April 2011, pp. 186 – 195.

[17] W. van Willigen, L. Kester, E. van Nunen, and E. Haasdijk, "Safety in the face of uncertainty: Critical headway estimation in cooperative adaptive cruise control." *International Journal of ITS Research*, vol. 12, no. 2, May 2014.