Bridging the Gap between Information Need and Information Acquisition

L. Ronnie M. Johansson
The Royal Institute of Technology (KTH)
Nada, Centre for Autonomous Systems (CAS)
SE-100 44 Stockholm
SWEDEN
`rjo@nada.kth.se`

Robert Suzić
The Swedish Defence Research Agency (FOI)
Department of Systems Modeling
SE-172 90 Stockholm
SWEDEN
`robert.suzic@foi.se`

Abstract – *In this article, we address the rarely discussed problem of connecting high-level information (e.g., aggregated states and enemy intentions) to information acquisition. Our approach is to partition the transition of information need to sensor management into a set of comprehensible entities (information types and functions), which we present in a framework. The framework is stepwise (sequential) and first translates actual information (from the data and information fusion process) to information need. The information need is mapped to the task space by a task management function which performs prioritization with respect to information need. A further step includes projection of tasks to service space by an allocation scheme, and finally services give orders to resources. In the terminology of the framework, we discuss the extension of a previous study (that involved plan recognition) with a sensor management function.*

Keywords: Sensor management, high-level information, situation and threat assessment.

1 Introduction

A set of methods that improve the process of collecting and reasoning about an uncertain environment (primarily through combination of information from disparate sources) is called *data* or *information fusion*. The goal is to describe a particular state of the world of interest by using all available information.

We consider the data fusion process (described in, e.g., [1]) to be a component of an enclosing system. We, furthermore, consider the data fusion component to support the enclosing system which is dependent on relevant external information to satisfy its objectives (e.g., to avoid being detected, disable enemy resources, or score in robot soccer, depending on application). The support provided by the data fusion process includes fusion of information from different sources and control of sensing resources. This idea is illustrated with the simple generic agent model in Figure 1, where *resources*, *knowledge*, and *control* and *fusion processes* may be spatially distributed. The arrows inside the system in the figure indicate control dependencies between the constituents. For instance, the data fusion process controls (sensing) resources and the system objective control can control and interfere with operation of the data fusion process. The arrows between resources and the environment indicate that the environment can be sensed and acted upon by resources.

Resources include sensors (e.g., radars, cameras and sonar) and effectors (i.e., resources, such as tanks in a command and control system or a manipulator arm on a mobile robot, that the system employs to achieve its objectives). Knowledge represents the long-term-memory of the system, possibly including facts and dynamic models of objects that can be observed in the environment. The system objectives control makes sure that effector resources are used to pursue the goals of the system.[1] The data fusion process combines information from sensors, control sensors to acquire relevant information, and possibly accesses and updates the knowledge base.

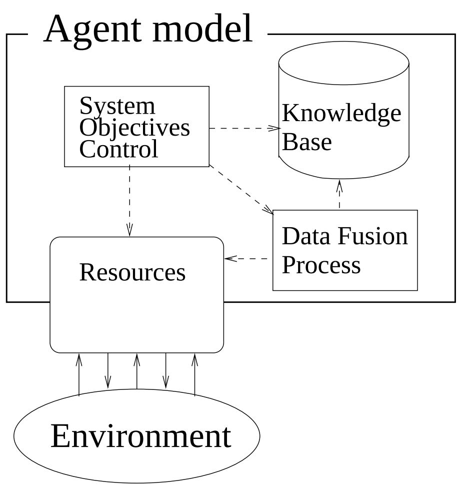

Figure 1: Data fusion in an agent context.

Efficient usage of sensing resources by the data fusion process includes adapting it to changing information need and requirements, not limiting the usage to specific (predefined) tasks. Clever control of sensors to acquire system-relevant information serves the purpose of the *process refinement* function in the JDL data fusion model [1] and improves the work and outcome of the data fusion process.

As suggested in [2], there are two preconditions for control of sensors or *sensor management* (as it is often called in command and control). They are that the sensors must be

[1] It is called system objectives control, rather than simply system control because part of the control (i.e., the control of sensing and perception) is managed by the data fusion process.

agile, i.e., they must be controllable so that a set of possible observations can be changed. Also, there must be some *conflict* in the control, e.g., that the information need is greater than can be satisfied by the sensors. Frequently, this control has to be dealt with in an automated manner since human beings might be unable to handle some chores in command and control systems that might be very complex or require a very fast response.

Efforts for sensor management have been surveyed in, e.g., [3, 4, 5, 6]. In most of the literature, it appears that the objective is fixed and aimed at improving the performance of the sensor system for the particular objective, rather than improving the relevance of the acquired information for the enclosing system. For instance, in a target tracking application, the system objective is limited to target tracking, while in a comprehensive large-scale system the objective could vary. The resources that were devoted to tracking could be utilized in detecting targets in some area instead, if that was the system's new objective. Another common characteristic of most efforts is that the information required by the enclosing system relates to entities or phenomena that can be immediately observed.

Higher levels of data fusion in the JDL model are jointly called information fusion. In those levels, assessment of an agent (hostile) force's intention and prediction of threat is performed. We call this kind of information *high-level information*. As a first step to obtaining high-level information and assessing its information need, data fusion activities such as identification, association, and classification, have to be performed.

In the second step, when revealing agent plans (related literature include [7, 8, 9, 10]), a priori knowledge about the enemy has to be modeled.[11]

Finally, on-line automatic multi-agent plan recognition has to be performed as a part of threat analysis. For example, an enemy unit that is expected to attack is more important to closely observe than a unit that is only marching. Therefore, by making a qualified estimation of the enemy's plans, we are able to assess the importance of a certain unit; and by combining importance with information uncertainty about a particular unit, the focus of attention would be inferred in order to connect high-level information need to sensor management. A problem is that on high levels information need is more complex and should be refined to lower abstraction levels. Those lower levels are have atomic components and represent comprehensible tasks for the sensor management. Therefore, we see a need for an automatic approach to task decomposition, task prioritization considering focus of attention, and task assignment for sensor management.

The main purpose of this article is to connect high-level information need, in our case the intention of an observed adversary, to the control of the sensors that are supposed to acquire the desired information. The information need discussed in this article originates from a hierarchical knowledge representation of the state and intention of the adversary.[12] A secondary purpose is to develop a generic framework for expressing the transition from information need to sensor control. Such a framework should be flexible enough to express various kinds of sensor management problems, including those which have multiple conflicting objectives and information need on different levels of abstraction.

We try to make the problem of translating information need to sensor control manageable by representing need by *information tasks* and the control space by *services*. A similar approach is discussed in [13, 14, 15], where the sharing of sensing resources among a set of platforms (e.g., aircraft or tanks) is addressed. Assuming a focus on efficient interaction between sensor platforms and their operators, they propose a distributed sensor management architecture. Their architecture is appropriate for use in network centric warfare where information should be shared among sensor platforms. Operator-selected so called *sensor management policies* ensure that the sensing resources on a platform are not used by tasks on other platforms in a way that would obstruct the mission of the platform. In contrast, our article is focused on a generic framework of the transition from tasks to sensor control. Hence, we are not assuming the use of only distributed sensor platforms.

Another related approach is offered in [16], where a hierarchical architecture for fusion of information and decomposition and distribution of tasks is described. Whereas the focus there seems to be on the design of hierarchical data fusion and resource management systems, our aim is to target the significant transition from information need to information acquisition.

Section 2 delineates the general framework for connecting information need to information acquisition which we propose in this article. Section 3 relates our framework to other structures proposed for sensor management and information acquisition. Section 4 suggests how we can use the framework to design a sensor management function for our application. Finally, in Section 5 we express our hope that this issue will receive more attention in the information fusion and sensor management communities in the future.

2 A framework connecting information acquisition tasks to sensing resources

In this section, we present a framework that emphasizes key issues in modeling and implementing the part of the data fusion process (JDL level 4) that connects information need to the acquisition of information. The purpose of the framework is to simplify the design of sensor management functions.

Our framework, depicted in Figure 2, prescribes that information need arising in some system (we call the source of this need the *task origin space*) is formulated as *information tasks* with assigned properties (e.g., priority or time horizon depending on what properties the system is designed to handle). Such tasks belong to the *task space* in Figure 2.

The materialization of tasks from information need could be the responsibility of a *task creation and management* function. The *service space* contains services that the sensors in the *resource space* (independently or jointly) can perform. The benefit of utilizing these services is, subsequently, the (more) relevant data that eventually is returned to the data fusion process by employed sensors. The

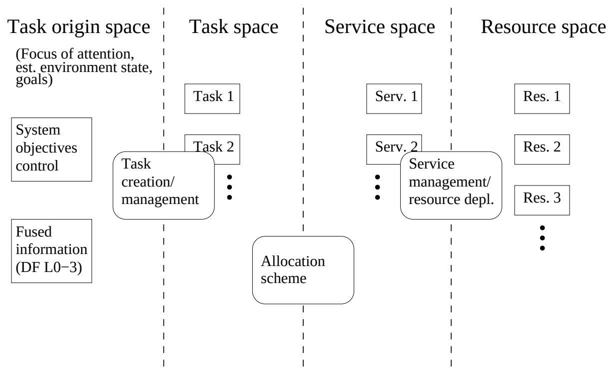

Figure 2: A framework for translating information need to sensor management

allocation scheme describes how tasks are connected to available services.

The general structure of the framework contains two types of entities: *space* and *function*. The four space entities: task origin, task, service and resource are containers of structured information. The structure of information of each space entity is amenable to the treatment of the intersecting function entities: task creation and management, allocation scheme, and service management and resource allocation. The purpose of the function entity is to convey information between its adjacent space entities.

Note that this framework does not suggest that the bridging procedure (from information need to its acquisition) is centralized in any way; tasks and services might be distributed and maintained separately (and are perhaps only made available locally) and the allocation scheme might be decentralized as well.

The tentative ontology, expressed in UML, of the framework shown in Figure 3 may further clarify the relations between the framework entities.

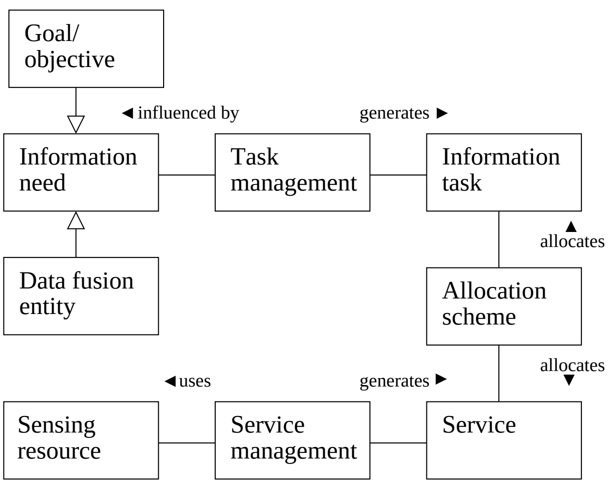

Figure 3: The ontology expresses the relations between the framework's space and function entities.

The following subsections will describe the different parts of the framework in Figure 2 in more detail.

2.1 Task origin space

We project information need to the *task space*, where tasks represent requests for information useful for a successful operation of the enclosing system. The creation of information tasks is spurred by the goals and focus of attention of the system objectives control and by the data fusion process itself. Thus, the character of the information tasks that emerge is influenced by the current information need of the system and demands of ongoing fusion processes.

2.2 Task space

The task space contains the information tasks spawned by the system's desire for more information (explained in the previous section). Describing relationships between tasks might facilitate efficient information acquisition. For instance, different information tasks might overlap, in which case that relationship should be noted. Tasks, typically, also have various attributes that are discussed in more detail in the next section.

2.3 Task creation and management

Given the contents of the task origin space, the task creation and management function forms information tasks that can be compared both to services and other tasks. The structure and information content of tasks is heavily dependent on the allocation scheme used (see Section 2.5). The structure is defined by various attributes such as *priority* (e.g., a value reflecting the importance of the tasks), *deadline* (a time point after which the tasks is irrelevant), *duration* (how long a continuous task should be served), and *origin* (the process which requested the information). Tasks could also express whether they must be solved completely (with a least quality requirement) or if partial solutions are feasible.

Overlapping tasks, i.e., that has some *commonality*, could be marked as dependent if the system wants to benefit from serving multiple tasks simultaneously. Task decomposition may be performed, e.g., to compare tasks and to find commonalities.

Note that the structure of tasks does not have to be identical for all tasks. For instance, tasks might be treated locally in clusters, where each cluster might have its own task structure.

2.4 Service space

Rather than connecting tasks directly to sensing resources, we propose an intermediate layer of services. There are several reasons for this approach. First, control of a resource is often abstracted, i.e., it is achieved through a set of modes which hides underlying design trade-off and optimization (design choices which could not efficiently be improved by a sensor manager).[2] Hence, control of a resource often involves initiating some process that manages the detailed operations of the resource. The service concept encapsulates both resource and detailed control.

Second, a sensing resource may provide several modes of operation, and rather than considering the resource as a single resource, we can consider the palette of modes, or services, that it provides.

Third, different constellations of sensors may form to provide services. Every such constellation and attached sensing activity could be considered as a service. There are numerous examples of previous efforts that group resources and treat them abstractly in this way [17, 18].

Consequently, the service space (ideally) contains the complete action space of sensor management. The space should only contain *feasible* services which can be achieved given available resources. It could possibly also contain services that are currently unachievable, but which are known to be achievable at a given moment or during a predictable time interval. The relation between resources and services is outlined in Figure 4.

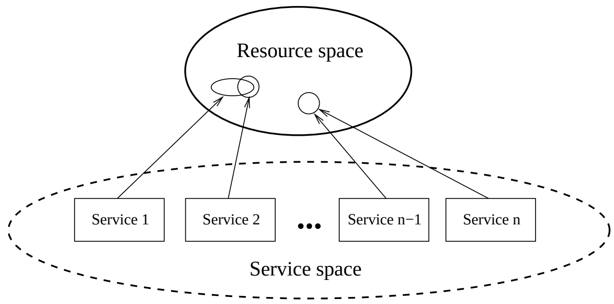

Figure 4: Individual sensors, sensor modes or sets of sensors constitute services.

Similarly to tasks, relationships between sensors may be expressed. Particularly, interference dependencies are of the essence. For instance, the operations of two services may interfere with each other by one emitting energy that will preclude the operations of the other. Furthermore, the engagement of a service may disable another candidate service since the resources that made it feasible are preoccupied with the engaged service.

2.5 Allocation scheme

It is the responsibility of the allocation scheme to connect information tasks to feasible services. The result is typically also a configuration specification for selected services. It is clear that design of this function for a particular application is heavily dependent on the structure of tasks and services.

The allocation scheme (i.e., the assignment of tasks to services) may appear in different guises. It may, for example, be a centralized scheme that collects all tasks and performs an exhaustive search over all combinations of tasks and services. A drastically different approach is to design tasks and services as agents which negotiate over allocations in a decentralized manner.

This function also performs re-prioritization of tasks. The reason for reconsidering the prioritization conceived by the task management function (in Section 2.3) is that high priority tasks, e.g., might not be satisfiable in the near future given the available services. In that case, a task which originally was assigned a high priority by task management has to yield in favor of low priority, but satisfiable, tasks.

2.6 Service management and resource deployment

The function of service management and resource deployment has two main chores: populating the service space with feasible services, and making sure that resources fulfill their obligations towards initiated services.

Feasible services are, as previously stated, services which can be achieved with available resources. Services which are probabilistically feasible (based on the system's belief in their availability and reliability) are also conceivable. Note that flexible sensing resources may concurrently serve multiple services, e.g., by reserving parts of their efforts for different services.

3 Framework compared to other structures

In this section, we relate the framework presented in the previous section to other known structures of sensor management and information acquisition: centralized/decentralized control, hierarchical layering, and the Macro/micro architecture.

The purpose of this article is not to replace the above-mentioned structures. Rather, here we discuss the compatibility of our framework with those structures.

3.1 Centralized and decentralized control

A fundamental issue when designing a sensor management system is whether the control should be *centralized* or *decentralized*.

A system that has centralized control has all decision-making (about sensor control) concentrated in a single computational node. Such a node, e.g., constituting a single CPU and memory, has access to all the information the system as a whole possesses. A prominent advantage of centralized control is, thus, that decisions about sensor control will be based on all of the information available to the system. Hence, the result is coherent (i.e., coordinated) sensing actions. Decentralized control does not enjoy this advantage; rather, decisions are based on information about the local environment and possibly rough descriptions of the global environment. Decentralized control, on the other hand, has other advantages such as robustness and scalability.

Many efforts in sensor management has assumed centralized control, perhaps because the focus in that line of research has been on optimal solutions rather than on control structures. Decentralized control is, however, highly relevant for decentralized fusion and the emerging field of network centric warfare (NCW),[19] and will likely receive more attention in the future. In NCW, sensors are assumed to form extensive networks and share information among themselves. In suchlike applications, the centralized control approach is simply unreasonable due to its lack of scalability.

Both centralized and decentralized control can be expressed in our framework. In the case of centralized control, typically, tasks are created and managed in a centralized fashion. The allocation scheme is also centralized. It uses the knowledge the system has acquired to consider all possible sensing actions (or more frequently, for complexity reasons, the optimal action is approximated through a search in the space of coherent sensing actions). Resources are inherently distributed (unless the application concerns only a single platform), but the attached service representation and management reside in the centralized node. Efforts that employ a centralized control include [20] (sensor placement) and [21] (target-tracking). In both of these articles, the task space only contains one fixed task. In the former article, it is the task of achieving the best detection probability, and in the latter that of the best expected improvement of the target state estimate.

In systems with decentralized control, tasks are created and maintained locally in distributed information nodes (i.e., nodes which require information for various missions). Likewise, services are created and maintained in sensor nodes. Hence, the allocation scheme will here have to find a suitable allocation and configuration of services that are distributed. One means to handle this problem is the contract net protocol (see, e.g., [22]) which represents tasks and service as agents. Task agents (*managers*) announce tasks to be solved and service agents respond with bids reflecting the cost (expressed in, e.g., time or energy) the enclosing system will suffer if selecting the service.

An example with decentralized control is [23] where a multi-target tracking problem for decentralized fusion on multiple platforms is addressed. Expressed in our framework, there is only really one task (which also is constant): maximizing expected measurement accuracy on the system on all targets. The services are the platforms themselves which each can take measurements of one target at a time. For complexity reasons, the allocation scheme in this case approximates the optimal solution. The resulting allocation of sensor platforms to targets is achieved after some cooperative work performed by the sensor platforms.

3.2 Hierarchical layering

The hierarchical *fusion and management dual node network* (FMDNN) in [16] is an architecture that tightly couples data fusion to resource management (a superset of sensor management). The FMDNN is expressive enough to capture both the design and operation of a data fusion system. Thus, the purpose of the FMDNN is different from the one of our framework. Whereas the FMDNN aids the development of a data fusion and resource management system, the purpose of our framework is to highlight the transition from information need to information acquisition.

3.3 Macro/micro architecture

The Macro/micro architecture [2] is a hierarchical sensor management architecture with two layers. It distinguishes two levels of functionality: the macro level which manages the overall information need and coordination of sensors (involving, e.g., distribution of tasks to sensors), and the micro level which deals with how a particular task is best accomplished by a particular sensor. The authors of [2] suggest that the sensor management function is divided into a centralized macro sensor manager (that enjoys the benefits of global information and coherent sensor control) and several sensor located micro sensor managers.

Our framework effectively encompasses the Macro/micro architecture. The responsibilities of the "macro manager" (e.g., task prioritization and scheduling) belongs to the task management and allocation scheme functions of our framework. The micro managers can be considered to be services (that encapsulates both sensors and tailored control processes) that the macro manager accesses. The sensor located services (i.e., micro managers) naturally makes sure that the underlying sensing resources perform the assigned task. Hence, the micro managers perform the work of the service management and resource deployment function.

Whereas the Macro/micro architecture is designed to support the common situation that a set of sensors on a single platform has to be managed, our framework is less to restrained (to encompass also, e.g., decentralized control).

4 Connecting high-level information need to sensing actions

In [12], uncertain sensor information, terrain information, and uncertain a priori knowledge about the enemy are inferred obtaining an estimation of enemy plans on different abstraction levels. We utilize a hierarchical Dynamic Bayesian Network (DBN) model (see, e.g., [24]) for this purpose. The methodology combines a set of heuristic rules that present sensor data in a soft manner to the DBN. The model represents estimation of plans as a discrete distribution of each enemy unit on each abstraction level. On platoon level, e.g., the plan space is: attack, defend, reconnaissance and march.

In Figure 5, we present a simplified DBN that has been used for plan recognition. This model follows military hierarchy by observing actions of tanks at lowest level inferring such information with knowledge of military organization and environment properties such as maneuverability, cover and weather.

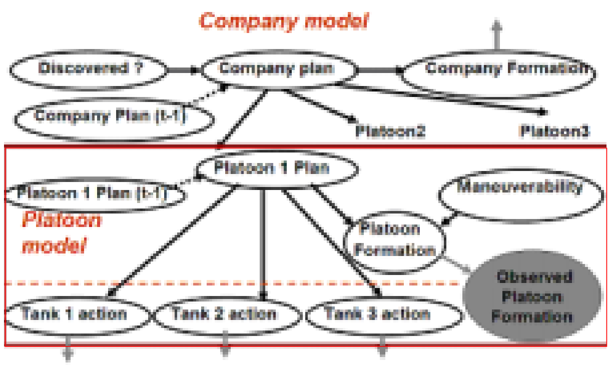

Figure 5: A simplified (Two Slice) DBN Used for plan recognition.

To extend our previous efforts, we here connect information need to sensor management by using a methodology that includes the information acquisition framework from Section 2. In this example, we use an an agent-based approach. We define the high-level information representation for situation and threat assessment (i.e., JDL levels two and three) as a set of agents. We see parts of the DBN from Figure 5 as agents. Each platoon model is an agent that communicates by using statistical inference and is aimed to present fused information. Therefore, we call them *information agents*. They represent a certain unit or phenomenon and its corresponding *information supplier agent* that has the responsibility for delivering surveillance data. Enemy plans, in this case, represent an activity that can be connected to importance for observing a certain unit. This connection is a part of the activity-prioritization process performed by an information supplier agent (see Figure 6). Consequently, we obtain linkage between high-level information such as activity and the need (priority level) thereof. An information supplier agent expresses information need as priority that in task space is translated into a set of atomic tasks that are further mapped to the service space.

Since the assumption is that service and resource management have limited capabilities the process of information supplier agents' prioritization and the prioritization of corresponding tasks has to be performed before assigning them to resource management. Prioritization of information acquisition tasks depends on which unit the information agent and its supplier represent.

However, the prioritization of tasks does not ultimately depend only on estimated plans. It depends as well on information uncertainty. For this application, we propose an approach to measure how inferred information is sensitive to changes of underlying information. In other words we pose an issue, is underlying information considering its uncertainty reliable and sufficient to infer robust a conclusion? If diminishing large uncertainty in underlying information causes small changes in the inferred result combined with low threat level, then we can say that large uncertainty is acceptable in this case. However, there are cases where large dispersion in inferred results is obtained from information with low uncertainty. Consequently, we say that this kind of underlying information which for small changes causes significant changes in result is more sensitive. To measure sensitivity, we propose to use a method that samples and calculates entropy measures of the inferred plans. If it turns out that estimations of plans are very sensitive to uncertainty of data in a particular situation then this information gets higher priority.

Finally, we state that prioritization of tasks is conditioned on both activity (subsequently expected impact; further discussed in [25]) and information sensitivity. The solution to automatic task prioritization could be modeled by using some soft computing method, i.e., a Bayesian network (BN) as in Figure 6 that takes as an input unit activity and unit uncertainty (sensitivity to uncertain information).

In Figure 7, the plan recognition algorithm assumes that the first and the last hostile platoon is attacking with very high probability (we are pretty certain) and information acquisition tasks for those two platoons should get high priority. At the same time, we are uncertain about the platoon in the middle and its intentions. Its estimated plan is sensitive to smaller changes. We do not know what this platoon is going to do. Finally, task prioritization use some a priori

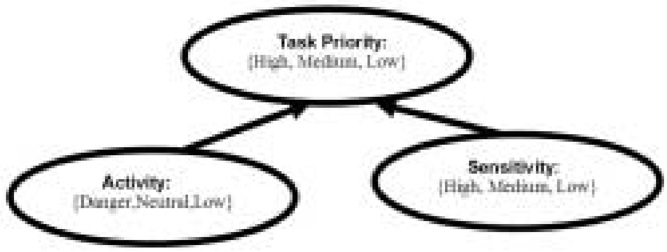

Figure 6: The task priority Bayesian network maintained by the information supplier agent.

rule, implemented in a BN in this case (Figure 6), to combine sensitivity and activity in order to get priorities that sensor management can use.

Figure 7: A hostile company (in the center of the figure with units moving in the directions of the arrows) and a friendly platoon on the right (from [12])

The other type of problem, when bridging from high-level information need to sensor management, is the task decomposition problem. There are different ways to express and decompose the information need into tasks. If there is a clear structure that assigns a task from higher levels and decomposes this to lower levels in order to reach a level of comprehensible tasks for service space, then we can say that we have a structure that decomposes high-context information and its corresponding information need to the task space. This decomposition task should be performed by some task decomposition scheme in the task creation. The relational structure of information agents can be used when composing a decomposition scheme. In our case, e.g., we have the relation that a platoon consists of three tanks, and this information about relations can be used when decomposing the task of getting the position of a platoon to subtasks of getting positions of tanks. It could, e.g., be the direction of motion of a platoon that is requested. Decomposition, with the help of the agent hierachy yields comprehensible tasks concerning the directions of the tanks that the platoon is composed of. A simple allocation scheme can now assign the tasks to available services (for instance the UAV with camera if the returned information is expected to reveal the direction of the tanks). If there are more tasks in the task space, their individual priorities will decide which tasks will be served first.

This task decomposition scheme takes into account prioritization of higher tasks and propagates it to lower levels. In some cases, tasks are overlapping, and we say that they have some commonality. By solving one task, a part of the other task might be solved simultaneously. Tasks that have been assigned priority and commonality, and that are refined to comprehensible level to services are mapped to service space. Two examples of services are image service and signal intelligence service. By taking commonality of tasks into account, in some cases, one service can satisfy more than one task. For example, taking a picture of a certain area that can be useful for multiple tasks.

Finally, services have to use sensor resources in order to fulfill requirements by tasks management. For instance, a camera mounted on a UAV (unmanned aerial vehicle) can be used for satisfying image service.

5 Discussion and future work

The challenging task of connecting high-level information fusion and sensor management should be studied further. In our article, we demonstrate how the concept of using a generic framework partitions and models the bridging process from information fusion to sensors management.

We have constructed the framework to encompass as much as possible of the sensor management concept. Yet, there are a few aspects of sensor management that are difficult to capture with our framework. For instance, decomposition of information acquisition tasks appears to be more complex than task decomposition in general. When decomposing information acquisition tasks, one has to make sure that the resulting, acquired information can be treated appropriately by the (possibly decentralized) data fusion process. Hence, the decomposition of tasks is dependent on the available fusion processes. We currently do not try to address this problem in the framework.

In the example concerning high-level information in Section 4, we suggest that both the agent (enemy) activity and its sensitivity to underlying information should have an impact on the prioritization of tasks.

In the case of [12], the use of knowledge about hierarchical relations between information agents can be used. When measuring sensitivity of position or direction of a platoon, e.g., aggregated data of the platoon's position and mean direction with standard deviation should be taken into account, not the positions and directions of each single platoon unit.

Also, in the example, we represent parts of the components with agents. We could bring this decentralized (multi-agent system) idea even further and introduce service agents and resource agents. Service agents would have the natural responsibilities of checking whether it is feasible (i.e., if there are resources available to make it achievable), and respond to requests posed by task agents. Resource agents could assess the status (e.g., remaining battery power and whether the resource is currently available) of the corresponding resource.

As mentioned in Section 4, the knowledge about relations between information agents can be used for decomposing tasks. This kind of approach enables modeling a flexible structure that performs mapping from the information agents hierarchy to the task decomposition structure.

A good cooperation with domain experts should be established when performing bridge modeling. The general framework represents a transparent and easily understood workflow process. It is easier to model and modify such a framework when bearing in mind clear distinctions between different functions of the transition from information need to information acquisition.

To date, only a few studies have been performed in this field. In fact, most studies treat the transition from information need to acquisition only implicitly. Moreover, the transition is often simplified by static task sets, low-level information need, and homogeneous sensors. We hope that this paper will arouse increased interest in this subject.

Acknowledgments

This work was financially supported by the Swedish Defence Research Agency (FOI), Division of Command and Control. The authors also wish to thank Pontus Svenson, Stefan Arnborg, Joel Brynielsson and Sandra Brunsberg for proofreading and advice prior to the submission of the paper. The progress of this work was, furthermore, stimulated by the meetings of the Decision Support Group.[2]

References

[1] Alan N. Steinberg and Christopher L. Bowman. Handbook of multisensor data fusion. In Hall and Llinas [26], chapter 2.

[2] Samuel Blackman and Robert Popoli. *Design and Analysis of Modern Tracking Systems*, chapter 15, pages 967–1068. Artech House, Norwood, MA, 1999.

[3] L. Ronnie M. Johansson. Information acquisition in data fusion systems. Licentiate Thesis TRITA-NA-03-28. ISSN 0348-2952. ISRN KTH/NA/R–03/28–SE. ISBN 91-7283-55-5. CVAP-283, Royal Institute of Technology, Numerical Analysis and Computer Science, Computer Vision and Active Perception Laboratory, 100 44 Stockholm, SWEDEN, November 2003. http://www.nada.kth.se/cvap/cvaplop/lop-cvap.html.

[2] `http://www.nada.kth.se/theory/dsg`

[4] Ning Xiong and Per Svensson. Sensor management for information fusion - issues and approaches. *Information Fusion*, 3:163–186, 2002.

[5] G. W. Ng and K. H. Ng. Sensor management - what, why and how. *Information Fusion*, 1:67–75, 2000.

[6] Stan Musick and Raj Malhotra. Chasing the elusive sensor manager. In *Proceedings of the National Conference on Aerospace and Electronics*, pages 606–613. IEEE, 1994.

[7] A. R. Pearce, C. A. Heinze, and S. Goss. Enabling perception for plan recognition in multi-agent air-mission simulations. In *Proceedings of the 4th International Conference on Multi-Agent Systems*, pages 427–428, 2000.

[8] H. H. Bui, S. Venkatesh, and G. West. Policy recognition in the abstract hidden markov model. *Journal of AI Research*, 17:451–499, 2002.

[9] Robert P. Goldman, Christopher W. Geib, and Christopher A. Miller. New model of plan recognition. In Laskey K. B. and Prade H., editors, *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, 1999.

[10] Frank Mulder and Frans Voorbraak. A formal description of tactical plan recognition. *Information Fusion*, 4:47–61, 2003.

[11] Robert Suzić. Generic representation of military organisation and military behaviour: UML and Bayesian networks. In *Proceedings of the NATO RTO Symposium on C3I and M&S Interoperability*, October 2003.

[12] Robert Suzić. Representation and recognition of uncertain enemy policies using statistical models. In *Proceedings of the NATO RTO Symposium on Military Data and Information Fusion*, October 2003.

[13] Fredrik Lantz and Dan Strömberg. Task management in sensor-provided operator platforms. pages 596–601.

[14] Dan Strömberg and Fredrik Lantz. Operator control of shared resources in sensor networks. In *Proceedings of the 6th International Conference on Information Fusion*, pages 575–582. International Society of Information Fusion, 2003.

[15] Dan Strömberg, Maria Andersson, and Fredrik Lantz. On platform-based sensor management. In *Proceedings of the 5th International Conference on Information Fusion*, pages 600–607. International Society of Information Fusion, 2002.

[16] Christopher L. Bowman and Alan N. Steinberg. Handbook of multisensor data fusion. In Hall and Llinas [26], chapter 16, pages 16–1 – 16–39.

[17] Patrick Muir. A virtual sensor approach to robot kinematic identification. In *Proceedings of the 1990 IEEE International Conference on Systems Engineering*, pages 440–445. IEEE, August 1990.

[18] Tom Henderson and Esther Shilcrat. Logical sensor systems. *Journal of Robotic Systems*, 1:169–193, 1984.

[19] David S. Alberts, John J. Garstka, and Frederick P. Stein. *Network Centric Warfare: Developing and Leveraging Information Superiority*. CCRP Publication Series, 2 edition, 1999.

[20] Dawn E. Penny. The automatic management of multi-sensor systems. In *Proceedings of the International Conference on Information Fusion*, pages 748–755, 1998.

[21] Wayne Schmaedeke. Information based sensor management. In *Proceedings of SPIE, Signal processing, sensor fusion, and target recognition II*, pages 156–164. SPIE, April 1993.

[22] Michael N. Huhns and Larry M. Stephens. Multiagent systems - a modern approach to distributed artificial intelligence. In Gerhard I. Weiss, editor, *Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence*, chapter 2, pages 79–120. MIT Press, 1999.

[23] Pierre Dodin and Vincent Nimier. Distributed resource allocation under communication constraints. In *Proceedings of the 4th International Conference on Information Fusion*. International Society of Information Fusion, 2001.

[24] Kevin Patrick Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, Universtiy of California, Berkeley, 2002.

[25] Stefan Arnborg and Joel Brynielsson. Bayesian games for threat prediction and situation analysis. In *Proceedings of the 7th International Conference on Information Fusion*. International Society of Information Fusion, 2004.

[26] David L. Hall and James Llinas, editors. *Handbook of Multisensor Data Fusion*. CRC Press, 2001.