14th International Conference on Information Fusion
Chicago, Illinois, USA, July 5-8, 2011

System Design for Distributed Adaptive Observation Systems

Maarten Ditzel, Leon Kester, Sebastiaan van den Broek
TNO, Oude Waalsdorperweg 63, 2597 AK The Hague, The Netherlands
Email: maarten.ditzel@tno.nl

***Abstract*—Currently, there is no clear-cut approach or design methodology available for designing distributed adaptive observation systems, partly due to the necessity to combine elements and approaches from several technological and scientific communities. Recently, an effort was made addressing this issue in an integrated manner, resulting in the definition of a model for Networked Adaptive Interactive Hybrid Systems, or NAIHS. However, so far its use was only demonstrated to a limited extent. This paper presents the application of the NAIHS model to design and implement four different prototypical fusion systems to show the applicability and relevance of the model's concepts.**

***Index Terms*—system design; system architecture; system modeling; distributed fusion; autonomous systems; fusion management**

I. Introduction

Today, when designing adaptive and distributed information fusion systems, there is no clear-cut approach or design methodology available. For one, this is due to the necessity to combine elements and approaches from several technological and scientific communities. For example, apart from the fusion domain, one can take methods and views from the systems engineering community [1]. However, these generally do not provide a system designer with a clear answer where to put (sub)system boundaries and interactions, as the applied methods typically are (and should be) application agnostic. On the other hand, insights from the wireless (sensor) networking community, might give the designer several approaches for the robust transmission of data over unreliable communication channels [2], but these will generally not address what information should be exchanged between parts of the system to be designed. Moreover, with the advent of autonomous systems, topics traditionally addressed during the design-time of the system, now need to be addressed at run-time, which requires new functionality to be added to an already complex system. Finally, yet another complicating factor is the use of different conceptions and terminology due to the inherent differences in historic development of the applicable communities.

Recently, an effort was made addressing these issues in an integrated manner, resulting in the definition of a model for Networked Adaptive Interactive Hybrid Systems, or NAIHS in short [3]. However, its use was so far only demonstrated to a limited extent [4]. This paper addresses the application of the NAIHS concepts to the implementation of several prototypical fusion systems. The systems vary with respect to the application domain, their targeted functionality, and their design constraints. Nevertheless, the same conceptual modeling is applied building from the same NAIHS concepts. Apart from the the successful implementation of these experimental fusion systems, it shows the relevance and applicability of the NAIHS concepts to the design and implementation of distributed and adaptive observation systems.

The paper is organized as follows. In Section II, an overview is given of the main NAIHS concepts. Then, in Section III to Section VI, four different projects are discussed, each targeting the implementation of a specific fusion engine. Finally, in Section VII we conclude the paper.

II. NAIHS Conceptual Modeling

The NAIHS model is based on a number of assumptions related to the design and operation of distributed observation systems. First and foremost, it explicitly takes into account that gathering and fusing information in an observation system is never a purpose per se. The information is intended to be used in some context, and in that context the value of the information is determined. Often, it is encountered that the context in which the information is used is only taken into account implicitly. The advantage of explicitly taking the information's use into account, is that knowing the value of information is an invaluable guide when making decisions on aspects like requirements definition, logical decomposition, resource allocation, system realization and optimization [5].

Second, NAIHS recognizes the fact that when targeting autonomous systems (for instance systems capable to adapt to changing environmental conditions, user requests, or the deterioration or failure of subsystems), processes traditionally executed at design-time, may now be required at run-time. As a consequence, extra functionality has to be added to the system, e.g., composition and monitoring functions.

The NAIHS model recognizes three different modeling principles to reason about design partitioning. These are (i) information abstraction, (ii) space-time abstraction, and (iii) physical structure. These are detailed in the next subsections. Thereafter, service composition is discussed, including a description of NAIHS' use of the notion of utility to optimize system performance.

A. Information Abstraction

The Information Abstraction classifies functionality based on its abstraction level. To that order, it borrows the layered approach from the well-known JDL model [6]. The JDL model identifies five levels of data processing and a database, all

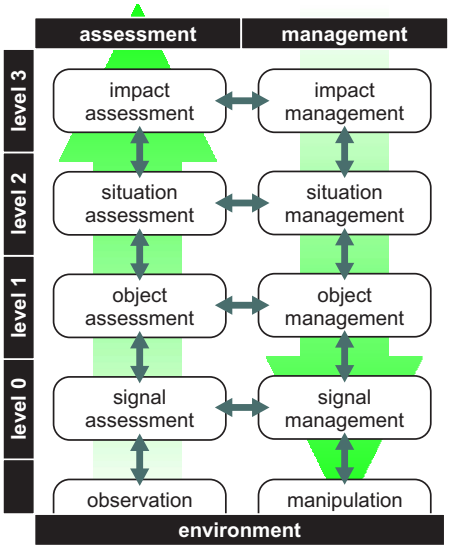

Figure 1. The NAIHS model uses a layered approach similar to the JDL model. Additionally, it differentiates between assessment and management functionality.

interconnected by a bus. From the JDL model the NAIHS model defines the following information abstraction levels (or layers):

- *Level 0 - Signal layer*
  The signal layer encompasses functionality related to the estimation of states of sub-object entities (e.g., signals and features);
- *Level 1 - Object layer*
  The object layer entails functionality acting at states of discrete physical entities (e.g., vehicles and buildings);
- *Level 2 - Situation layer*
  The situation layer contains functionality dealing with relationships between entities (e.g., grouping, cueing, approaching);
- *Level 3 - Impact layer*
  The impact layer addresses functionality related to the estimation and management of impacts (e.g., expected consequences of one's own or other one's – possibly hostile – activities).

Apart from the layers mentioned above, taking the OODA loop (Observe, Orient, Decide, Act) as example [7], functionality is also divided into distinct *assessment* and *management* categories, resulting in a symmetric model as can be seen in Figure 1 and is also recognized in [8]. The resulting figure shows a remarkable similarity to the hypothetical hierarchical structure of the human mind as presented in [9].

Assessment functions are concerned with the construction of an appropriate picture of the situation (and its impacts) at hand. This corresponds to OODA's observe and orient steps. Management functions implement decision making processes aiming to take adequate actions in response to arising situations and threats, which is in line with the decide and act steps of the OODA loop. Moreover, the management part explicitly deals with the management and control of the observation system's own assets and resources. It does so, in order to optimize the system's overall configuration to adapt to changing conditions and to optimally fulfill the system's goals.

B. Space-time Abstraction

Every decision making or management process reasons about possible outcomes of the future taking into account possible actions. The time horizon of these processes may have a wide range. Therefore, a decomposition of the process cycle in this dimension has been adopted in various application domains. In the military field a strategic, operational and tactical level is in use. For (business) planning the same levels are in use, however, contrary to the military case the operational level acts on a shorter time scale than the tactical level.

In the artificial intelligence domain, Brooks [10], who discards the decomposition in information abstraction, proposes a hierarchical composition of process cycles based on reaction or cycle time. The need for decomposition into temporal abstractions is also acknowledged in the case of decision making processes in complex situations. The time scale may differ widely for various applications. A suitable decomposition therefore depends on the application.

Most models use only one type of abstraction, either information or space-time. There are some that distinguish these different principles but integrate them in one abstraction hierarchy, see for example [11], [12]. Although it is a tempting thought to do so, in the NAIHS model we would like to consider the two principles on which decomposition can be based separately. The reason for this is that it is well imaginable that information processing at a high information abstraction level is very fast and relevant for short term decision making. Moreover, the information processing from one sensor at one platform needs to be decomposed in many components at the higher information levels. Furthermore, results presented in [13] indicate that both abstraction principles play a role in hierarchical behavior. In the functional model, therefore, the freedom is maintained of accommodating both abstractions.

C. Physical Structure

The third modeling principle of the NAIHS model is related to the physical structure. The distributed observation systems considered typically consist of several platforms (or nodes) with various collectors and effectors. For instance, a platform can be an unmanned aerial vehicle, carrying an air traffic radar and a ground-looking infrared camera. Each platform contains one or more resources that provide services which can be used to implement the system's desired functionality. As depicted in Figure 2, the identified resource services are:

1) *Observation* – The ability to observe the physical system and its (physical) environment, which is bounded by time, space and available spectrum;
2) *Manipulation* – The ability to make physical changes to the system and its environment;

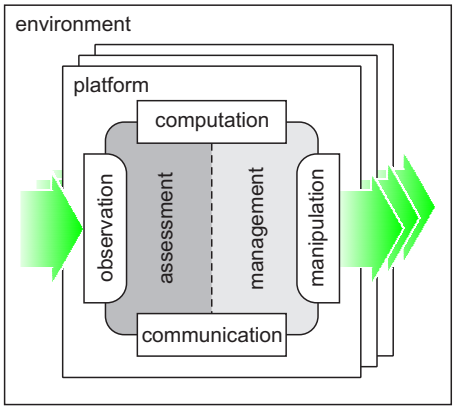

Figure 2. A system consists of platforms containing resources which provide computation, communication, observation and manipulation services. The observation and manipulation services are able to interact directly with the system's environment.

3) *Computation* – The ability to make calculations and run algorithms;
4) *Communication* – The ability to exchange information within the system and to communicate with external (other) systems.

Of course, all the aforementioned services require energy to operate. However, for clarity, the use of energy is not explicitly addressed in this paper.

D. Service Composition

If one decomposes the system's functionality according to information abstraction and time horizon, apart from its physical structure – as done in the NAIHS model – one has to address their couplings, to come to a working system, as they are by no means independent. This automatically leads to the topic of service composition: the interaction and organization of the offered services to come to the system's overall functionality. In the NAIHS model, service composition can be performed both at design-time or at run-time (or any mixture). The designer has to carefully judge *where* and *when* to put this functionality. As stated earlier, the foreseen use of the information generated by a service composition will guide this decision.

Figure 3 gives an example of a mapping of functions to platforms. In this figure, the physical structure can be seen adding a third axis to the originally two dimensional abstraction plane of Figure 1. Note that there is no one-to-one mapping of functions to platforms. Assessment and management functions of every level can be mapped to distinct platforms.

A particular aspect on which NAIHS puts attention is run-time service optimization. In an a-priori unknown situation the functionality of the services cannot be fully determined at design-time. This results in changing requirements on quality and timeliness of services. This request can be expressed in the form of a utility function. Fulfilling the request requires a capability to reason about generating value, as defined by the utility function, at what computation, communication or information provision cost.

We want to stress here that it is not the object of system design to implement all possible capabilities the NAIHS model indicates, but to produce the design of the most efficient and effective system. We believe, however, that more conscious and therefore better design decisions can be made if the possible alternatives are clear. In general if certain capabilities are not expected to generate more value than the cost to produce them, they will not be developed.

III. Test case 1: DAFNE

At the end of 2006, the European Defence Agency (EDA) launched a large research program on Force Protection [14]. The primary goal of the program is to improve the protection of European armed forces against the threats the troops encounter in present-day battlefields, with a special focus on urban environments. One project currently executed in the program is DAFNE: *Distributed and Adaptive multi-sensor FusioN Engine*. The aim of DAFNE is to demonstrate the benefits of fusing data of multiple heterogeneous sensors combined with intelligence sources over single-sensor operations in an urban environment.

As an integral part of the project, a multi-target, multi-sensor and multi-platform fusion engine is developed and implemented. The engine should be capable of mission dependent tracking, classification, situation and threat assessment. The engine's data flow is depicted in the SysML diagram in Figure 4. The following components are identified: (i) the DAFNE Fusion Engine and (ii) the Fusion Management. These are discussed in more detail in the next two sections, followed by an exemplary scenario. The example highlights two aspects of the NAIHS model in particular: information abstraction and utility based service optimization using contextual information.

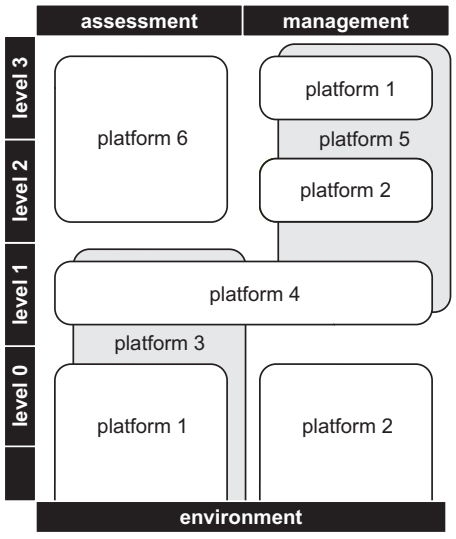

Figure 3. Example of realization of NAIHS functionality on different platforms. Note that similar functionality can be realized on different platforms.

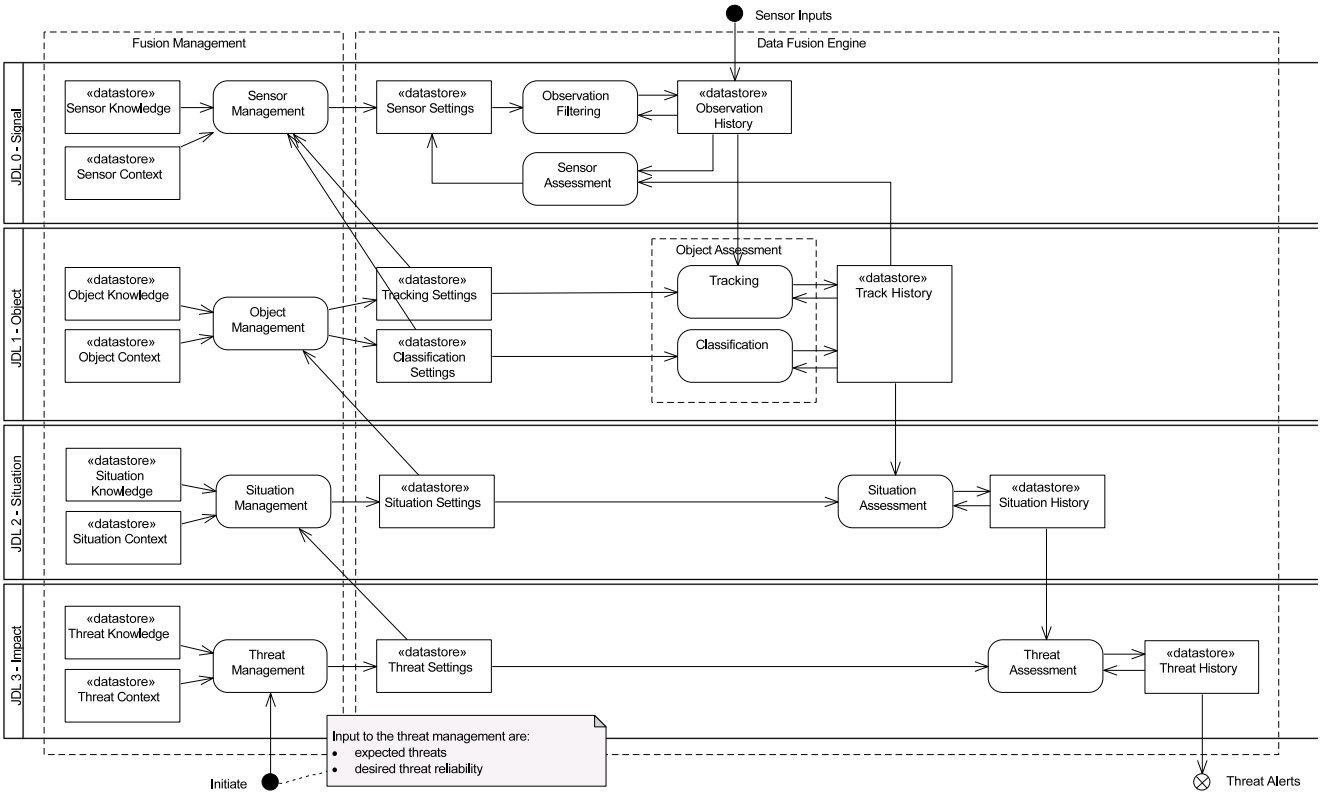

Figure 4. Data flow diagram of the experimental DAFNE fusion engine.

A. Fusion Engine

The fusion engine contains functionality related to the *assessment* (right) side of Figure 1. Input to the fusion engine are observations from multiple heterogeneous sensors. A layered approach is taken, resulting in signal, object, situation and threat assessment functions. Associated with each function are two types of datastores: one for holding the intermediate results per layer, and one for the context dependent settings. The main flow in the fusion engine is top-down (as depicted in Figure 4): signals lead to objects lead to situations lead to threats. The functions in this flow are optimized using contextual information provided by the functions' corresponding contextual settings.

B. Fusion Management

The fusion management generates the contextual settings for the fusion engine. As such, it implements the *management* (left) side of Figure 1, particularly focusing on self-management. Part of this process is service composition and optimization, constructing and configuring the fusion engine. Again, layering is applied to partition the functions. Input for each management function is the a-priori knowledge consisting of mission dependent context and mission independent knowledge. Mission dependent context may contain geographical information like maps and plans of the mission area. Mission independent knowledge contains less volatile information such as classification rules, etc.. The main flow in the fusion management is bottom-up (the reverse of the fusion engine): expected threats lead to relevant situations lead to objects and signals to look for.

C. Fusion Optimization

Apart from contextual management, the engine also contains a feedback mechanism to optimize the performance of the fusion engine. All observations from the sensors are initially stored in the *observation history* datastore. The *observation filtering* filters the data using initially only contextual information. For instance, an observation can be judged to be a false positive, using geographical information of the surroundings (e.g., the street plan of a city is used to discard false detections of a car inside a building, caused by reflections or other propagation effects). Then, the filtered observations are passed to the tracker. The results of the tracker are stored in the *track history* datastore, which together with previous tracks form the engine's best estimate of its environment. When the estimates have sufficient reliability, they are used to a-posteriori judge the validity and usefulness of the past observations. This may lead for instance to the exclusion of sensors which steadily provide erroneous observations, or even to the correction of errors such as a systematic offset of a sensor.

D. Scenario

To test the DAFNE fusion engine, the engine is subjected to several scenarios. One of the scenarios describes a possible terrorist attack on an embassy in a city. The purpose of

the DAFNE system is to autonomously combine the data of several sensors with intelligence data in order to alert security personnel of an imminent threat to the embassy.

In an accompanying paper [15], this scenario is discussed in more detail, focusing on the higher level situation and impact assessment and management functionality. The paper also presents the results of the application of the DAFNE fusion engine in the embassy scenario to detect threatening situations in an early stage. The presented results show the applicability and validity of the aforementioned approach to build an adaptive fusion engine.

IV. Test Case 2: Shared Picture Compilation

In a defense research program on picture compilation in sensor networks the aim is to generate shared awareness in a task group of frigates. Here, communication constraints limit the exchange of all valuable information. As a result, current solutions suffer from latencies in the compiled pictures and ambiguity in the tracks. It was understood that if the system functionality can reason on exchange of most valuable information under changing (possibly unexpected) situations, this could considerably improve system efficiency and effectivity. Several papers have already been published on the subject [16], [17], so here we limit ourselves to how NAIHS conceptual modeling was used for identifying task group level functions.

A. Task Group Level Functions

In the total system four task group level functions can be identified.

1) *Situation Awareness* – signal assessment and object assessment
2) *Threat Evaluation* – situation assessment and impact assessment
3) *Weapon Assignment* – impact management and situation management
4) *Engagement* – object management and signal management

This matches the OODA loop and also the NAIHS model of Figure 1 well. In addition, considering the resources depicted in Figure 2, specific resource management functions are needed for observation (sensors) and communication, since their dynamic allocation is crucial for an efficient and effective operation of the task group.

The importance of feedback loops between the task level functions was identified. Limited communication resources are most constraining for the 'Situation Awareness' function since this function requires most bandwidth at lowest latency. This means that in changing situations it is most important to have feedback from the 'Threat Evaluation' function on what information, indicated by the utility function, is most important.

B. Task Group 'Situation Awareness'

The situation awareness function can be decomposed in the signal assessment and object assessment function. The latter is further decomposed in tracking and classification functions. Currently, shared situation awareness in task groups is generated by exchanging (recognized) tracks through the Link-16 Tactical Data Link [18].

A disadvantage of this approach is that there are inherent latencies in the Link-16 communication system and ambiguities can result, due to individual tracking and recognition by each member of the task group and subsequent combination of this information. An obvious improvement is to use communication systems with lower latencies and, in order to prevent ambiguities, exchange measurements instead of recognized tracks. However, to exchange all measurements communication systems with low latencies and high bandwidth are needed, which can only be realized at tremendous cost.

Realizing that, according to the NAIHS model, any function can be implemented as a service, a way out of this dilemma is to implement the tracking algorithms as services. In addition to their tracking capabilities the services have to reason about the value of their tracking results, based on the (possibly changing) request of the task group's 'Threat evaluation' function.

The decision to exchange measurements is dependent on the 'expected reward' of communication of a particular measurement. This reward is equal to the added value of that particular measurement minus the expected cost of communication to the other members of the task group. Also, the capabilities of the communication system may change during operation. Therefore, communication could also be implemented as a service expressing the expected cost of communication to the tracking service. The mathematical method on how tracking services can reason about optimizing their service in interaction with a communication service is described in [17]. The communication service model is presented in an accompanying paper [19].

V. Test case 3: ICAR

A second project in the EDA Force Protection program mentioned above in Section III, is the ICAR project: *Intelligent Control of Adversary Radio Communications*. Again aiming to increase the survivability of European troops, it targets improved gathering and use of information, in this project focusing on electronic warfare sensors and effectors.

Figure 5 depicts the context diagram of the ICAR system. The main purpose of ICAR is to detect and if necessary selectively jam hostile transmitters such as cell phones and remote controls. It does so by sensing transmissions in the radio spectrum. The sensed data is analyzed and combined with information from other sources (modeled by the actor *External sensory information and intel*). The threat posed by the transmitters is assessed and, if necessary, appropriate countermeasures are taken, either automatically, or initiated by the *ICAR user*. Electronic warfare (EW) countermeasures may involve the selective jamming of the hostile transmitter, but also non-EW measures are possible (via the corresponding actor *Effectors (non-EW)*. Finally, also the actor *Platform control* is included, to steer the platforms carrying the ICAR sensory equipment (such as a UAV).

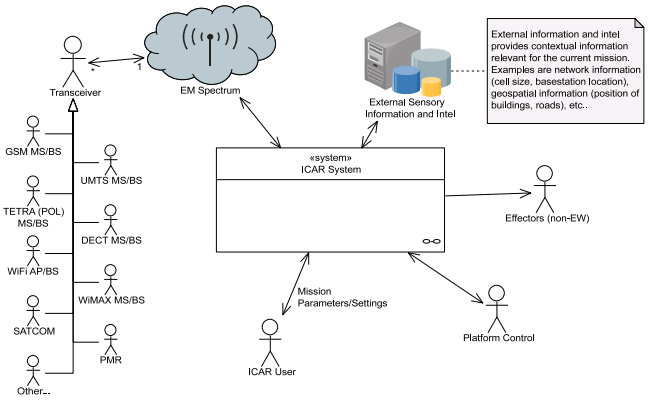

Figure 5. Context diagram of the ICAR system.

Internally, the ICAR system implements the core functionalities indicated in Figure 6. Now depicted horizontally, ICAR uses the same functional decomposition as described in Figure 1. ICAR's functionality is partitioned into the four information abstraction layers (vertical lanes: signal to impact), as well as separated into an assessment (upper half) and a management part (lower half). These are discussed in more detail in the next two sections.

A. Spectrum Assessment

Following the upper half of Figure 6, ICAR first senses the spectrum to find radio transmissions. Then, their characteristics are extracted and used to estimate the location of the transmitters. Together with information from other sources, the current situation is constructed and assessed and the threat imposed by the situation is analyzed.

A key aspect for the success of the ICAR system is the ability to properly distinguish between hostile and non-hostile transceivers. Especially in an urban environment many transceivers are active. Therefore, after activation, ICAR first attempts to construct the "normal" situation, which is then used to find any significant deviations (anomalies).

B. Spectrum Management

An important issue for the ICAR system is the ability to react quickly to imminent threats. For instance, in case of an RC-IED (radio-controlled improvised explosive device), there is little time to react between the remote activation of a device and its explosion. Current practice to continuously jam several suspect radio bands has several negative side effects, such as disturbance of own and public (non-hostile) communications, together with a prolonged exposure of the own troops to the emitted electromagnetic waves.

With the ICAR system, another approach can be taken. Given the (at run-time) estimated "normal" situation, one or more trigger masks (defined at design-time) can be selected and configured to trigger jamming for specific events. In case of the detection of an anomalous transmitter, the selected trigger mask(s) are used to quickly start instantaneous and selective jamming. The same procedure can be used, if external sources inform the ICAR system of an imminent threat (for instance a suspicious cell-phone). This mixture of design-time and run-time service composition enables selective yet fast responses.

VI. Test Case 4: Cortex

In the Cortex project, a Visual Intelligence (VI) system is being developed. This work is part of DARPA's "Mind's Eye" program [20]. In this program research is being done towards development of autonomous observation systems for recognizing action in a scene observed by camera systems, and reporting about the action. This action is defined around several verbs. In the near future, such systems would be deployed as on board intelligence for robotic observation platforms carrying cameras.

In the Cortex project, many possible ways of information extraction and reasoning will be looked at to create a system that is able to perform adequately on a wide range of scenes and situations. For this reason, the aim is a flexible fusion engine, which allows adding and/or modifying algorithms at different levels, and switching between them. The basic functional design is shown in Figure 7. Functionality is ordered according the NAIHS' levels of information abstraction, covering the signal, object and situation assessment and management (see Figure 1). Although it is a fairly complex design, the structured design shows the applicability of the NAIHS model to design larger systems.

A. Cortex system description

The Cortex VI system processes clips of video and produces reports about observed actions. Processing follows several functional modules, as shown in Figure 7, which are described in more detail below. The modules are ordered along information levels, starting with signal level video clips as input and ending with situation information as output that would be passed to overall situation and impact assessment.

1) Visual Processing: In the Visual Processing module, different algorithms are used to obtain information about possible entities. This includes feature extraction, detection (segmentation), classification and tracking. Once realized (physical structure), sub-modules may cover several of those at once. For example, a person detector may result in a detection with classification, and tracking may use information such as color, resulting in features linked to tracks. Output of sub-modules (such as person detection) may be used internally as (additional) input of other sub-modules (such as tracking). Output of this module is mostly object related, but also encompasses features that will be linked to objects at a later stage in the processing.

2) Entity Fusion: The Entity Fusion module continues with object assessment. It combines all visual information into descriptions of possibly relevant physical entities in time. These entities can be persons, cars, and other relevant objects. The entities are described in time as tracks annotated with additional information (such as classification).

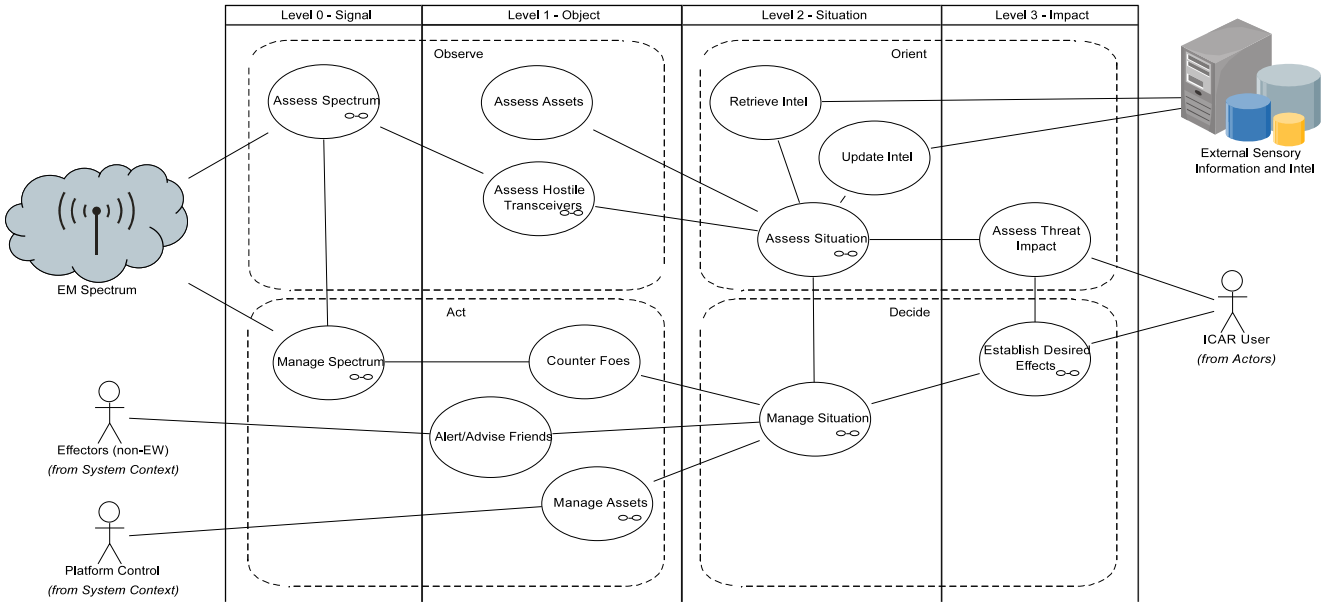

Figure 6. Use case diagram of the ICAR system.

Functionality in the Entity Fusion includes filtering and merging information. Clutter is reduced by filtering input information (detections and tracks). Tracks are merged if they are assessed to be describing the same entity. Other information of features, detections and classification are added to the descriptions based on location of the resulting entity description. Fusion also includes linking partial tracks in time, for example using features linked to both parts to recognize they are describing the same entity.

3) Event Description: The Event Description module both performs object assessment (in describing single physical entities in time) and situation assessment (in describing relations between entities). For example, a list is made of when an entity's location is going up or down, when it is moving (and if this is fast), when it has a certain shape, and if it matches certain requirements (e.g., is it a person). Additionally, relations are determined between entities. For instance, when the distance between entities is decreasing, increasing, small, or zero. All this information is stored on the Blackboard.

4) Blackboard: The Blackboard is mainly a data store for all information about the clip as determined by the previous modules. It does however also have limited functionality, in that selective information can be retrieved using dedicated requests.

5) Reasoning: Situation Assessment is continued in the Reasoning module. Different algorithms can be used to link the event descriptions to the defined verbs. One possibility is an expert system. In this case, new requests can be made to the blackboard based on the current state, such as "Are there two persons?" or "Is distance between two entities decreasing?". The result of Reasoning is an assessment for all defined verbs on whether it is likely that it is present in the action.

6) Anomaly Detection: Based on previous assessments on what verbs are normally observed, anomalies can be determined, and the observed action flagged as such.

7) Reporting: The observed situations (verbs and anomaly detection) are converted in requested formats, some of which are human readable (e.g., "Person in blue passes red car.").

B. Fusion Optimization

Parts of the VI system are tuned and trained for a known set of scenes. The aim is to make the system robust for use in different situations. During development, modules earlier in the chain are tuned based on the required information at higher level modules. For example, in Entity Fusion, clutter reduction is tuned to prevent flooding the Event Description module with too many entities. Also while developing, there are little restrictions on the amount of computation. For example, many features can be computed for whole images, selecting what is needed (near tracks) afterwards.

In a final system, computational limits will make this approach infeasible. Selective feature computation, based on the utility of information, is already partly implemented using a refinement feedback from the Entity Fusion module. Certain features are only where relevant entities are found, and only at times where this is assumed to add information needed in fusion and later modules. Another feedback loop is in the Reasoning module, asking the Blackboard only for information it needs. In a later implementation, this mechanism of optimization could be used to trigger a chain of requests down to lower levels, only fusing information for entities that are relevant for the requests made by the Reasoning module, and only determining classifications that are needed for those.

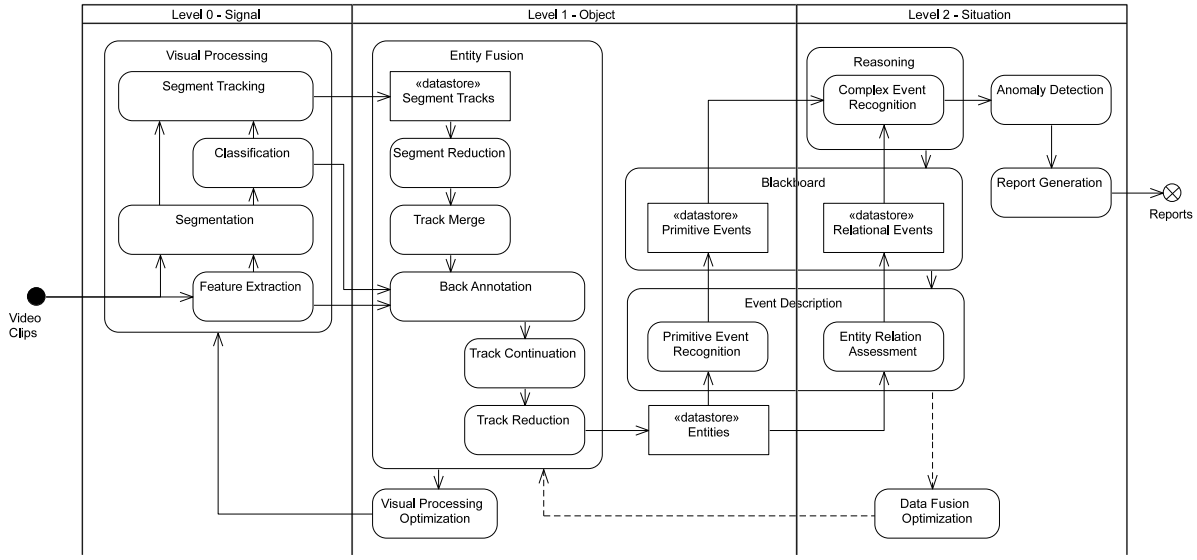

Figure 7. Functional description of the Cortex Visual Intelligence system.

VII. Conclusions

This paper has described the application of an integrated model (NAIHS) for designing distributed adaptive observation systems. The model is based on three modeling principles: information abstraction, space-time abstraction and physical structure. To guide design decisions and optimize system performance, the concept of the utility of information is applied, taking the use of generated information in its context explicitly into account.

So far, the NAIHS model's use had only been demonstrated to a limited extent. Therefore, we have presented four different designs of prototypical fusion engines which were developed using the NAIHS concepts. The successful application of the conceptual modeling shows the validity and relevance of the model.

Acknowledgment

Part of the presented work is done as part of the DAFNE and ICAR projects sponsored by the European Defence Agency (EDA) under grant numbers A-0380-RT-GC and A-0935-RT-GC. The Cortex project is sponsored by DARPA [20]. Moreover, the authors would like to thank Patrick Hanckmann, Miranda van Iersel and Johan-Martijn ten Hove for their valuable feedback and constructive criticism.

References

[1] K. J. Rothenhaus, J. B. Michael, and M.-T. Shing, "Architectural patterns and auto-fusion process for automated multisensor fusion in soa system-of-systems," *IEEE Systems Journal*, vol. 3, no. 3, pp. 304 – 316, September 2009.

[2] H. Luo, H. Tao, H. Ma, and S. K. Das, "Data fusion with desired reliability in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 3, pp. 501 – 513, March 2011.

[3] 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.

[4] L. Kester, "Networked adaptive interactive hybrid systems (NAIHS) for multiplatform engagement capability," in *European Radar Conference*, October 2008, pp. 308 – 311.

[5] National Aeronautics and Space Administration, *NASA Systems Engineering Handbook*, December 2007.

[6] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," in *Proceedings of the IEEE*, vol. 85, January 1997, pp. 6 – 23.

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

[8] N. Steinberg and C. Bowman, "Revisions to the JDL data fusion process model," in *Proceedings of the 1999 National Symposium on Sensor Data Fusion*, May 1999.

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

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

[11] C. B. Alan Steinberg, "Rethinking the JDL data fusion levels," in *Proceedings of the national symposium on Sensor Data Fusion JHUAPL*, June 2004.

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

[13] J. Rasmussen, "The role of hierarchical knowledge representation in decision making and system management," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 15, pp. 234–243, 1985.

[14] European Defence Agency , "EDA defence R&T joint investment programme on force protection (JIP-FP)," http://www.eda.europa.eu/genericitem.aspx?id=184.

[15] S. P. van den Broek, P. Hanckmann, and M. Ditzel, "Situation and threat assessment for urban scenarios in a distributed adaptive system," in *14th International Conference on Information Fusion*, 2011.

[16] M. van Iersel, L. Kester, J. Bergmans, P. Hiemstra, K. Benoist, and B. van den Broek, "Creating shared situation awareness in a multi-platform sensor network," in *11th International Conference on Information Fusion*, July 2008, pp. 1 – 6.

[17] 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.

[18] NATO Standardization Agency, *NATO Standardization Agreement (STANAG) 5516: Tactical Data Exchange – Link 16*.

[19] E. van Foeken and M. Kwakkernaat, "Low-complexity wireless communication modeling for information flow control in sensor networks," in *14th International Conference on Information Fusion*, 2011, pp. 118 – 125.

[20] DARPA , "DARPA mind's eye program)," http://www.darpa.mil/Our_Work/I2O/Programs/Minds_Eye.aspx.