Abstract – This paper reports on methods for the cost-effective development and integration of multi-sensor fusion technology. The methods presented extend the Project Correlation Data Fusion Engineering Guidelines with significant evolution. The key new insight is in formulating the system engineering process as a resource management problem; allowing the application of the Bowman’s model of the duality between data fusion and resource management.¹

Keywords: Data fusion, data association, estimation, system engineering methodology, problem decomposition, resource management, planning, information integration, information fusion, engineering guidelines, adaptive modeling, system acquisition, technology acquisition, scientific method, inference.

1 Data Fusion Engineering

1.1 Scope of Data Fusion

Following the 1998 revision to the JDL Model definition, given in [1], we define “data fusion” as a *process of combining data or information to estimate or predict entity states.*²

So-defined, data fusion pervades all biological cognitive activity and virtually every automated approach to the use of information. Unfortunately, the very universality of data fusion has engendered a profusion of overlapping research and development in many applications. A welter of confusing terminology and *ad hoc* methods in a great variety of scientific, engineering, management and educational disciplines obscures the fact that the same ground has been plowed repeatedly.

Often, the role of data fusion has been unduly restricted to a subset of the processes and relevant to particular state estimation problems.

For example, in military applications such as targeting or tactical intelligence, the focus is on estimating and predicting the state of specific types of entities in the external environment: targets, threats, military formations, etc. In this context, the applicable sensors/sources that the system designer considers are often restricted to the obvious offensive, defensive and/or surveillance sensors and other live sources.

Ultimately, however, such problems are inseparable from problems of navigation, of calibrating sensor alignment and performance, and of validating one’s library of target models. A more powerful realization of the role of data fusion – and, indeed, of resource management, as well – is one in which all sources are exploited to solve all required state estimation/prediction problems.

The estimate and prediction of states of targets, other external entities (threats, terrain, weather, etc.) – as well of one’s own platform, its sensors and other systems – is a single problem, amenable to a unified self-consistent solution. The evaluation of the system’s models of the characteristics and behavior of all of these external and organic entities is likewise a component of the single problem of estimating the actual world state.

¹ Acknowledgement – the author wishes to acknowledge the contribution of Dr. Christopher Bowman, both in developing several of the original concepts presented and in helpful critique of this document.

² It is fairly pointless to argue whether the term data fusion or some other term is an appropriate label for this very broad concept. There is no body of common and accepted usage to which we can appeal for such specialized terms. What is important is the recognition that this broad concept is an important topic for a unified theoretical approach, and therefore deserving of its own label. Some people have preferred terms like ‘information integration’ with an attempt at connoting greater generality than earlier, narrower definitions of data fusion (and, perhaps, to divorce oneself from old data fusion approaches and programs). There is danger, however, in neglecting relevant research by willful relabeling.
Correlation[3-5]. The Guidelines were developed as part of the U.S. Air Force Space Command’s Project Engineering Guidelines that were developed in 1995-96. The present work builds on a set of Data Fusion Project Correlation Data Fusion multi-spectral models of targets and collection systems.

The Guidelines recommend an architecture concept that represents data fusion systems as networks of processing nodes, each node having the structure shown in the upper half of Figure 2. When the data fusion process is partitioned into multiple processing nodes, the process is represented via a data fusion tree, illustrated in Figure 3.

Integral to the Guidelines was the use of a functional model for characterizing diverse system architectures and processing and control functions within a data fusion process. This architecture paradigm, deriving from a considerable body of work by Bowman[6], has been found to successfully capture the salient operating characteristics of the diversity of automatic and manual approaches that have been employed across great diversity of data fusion applications.

The Guidelines recommend a four-phase process for developing data fusion functionality within an information processing system, shown in Figure 4. Design and development decisions flow from overall system requirements and constraints to a specification of the role for data fusion within the system. Further partitioning results in a specification of a data fusion tree

1.2 Project Correlation Data Fusion Engineering Guidelines

The present work builds on a set of Data Fusion Engineering Guidelines that were developed in 1995-96 as part of the U.S. Air Force Space Command’s Project Correlation[3-5]. The Guidelines were developed to provide

- a standard model for representing the requirements, design and performance of data fusion systems and
- a methodology for developing multi-source data fusion systems, selecting among system architecture and technique alternatives for cost-effective satisfaction of system requirements.

If it can be assumed that the universe of discourse can be partitioned into a finite number of entities of interest, but that that number, \( k \), is unknown, then the problem of consistently estimating a multi-object world-state is shown in Figure 1 (based on [2]). Here, \( x_i, \ldots, x_k \) are entity states, so that the global state estimation problem becomes that of finding the finite random set \( X \) with maximum likelihood.

That the data fusion system engineering process is far from a trivial one is characterized by difficulties in

- representing the uncertainty in observations and in models of the phenomena that generate observations;
- combining non-commensurate information (e.g. the distinctive attributes in imagery, text, and signals);
- maintaining and manipulating the enormous number of alternative ways of associating and interpreting large numbers of observations of multiple entities.

Deriving general principles for developing and evaluating data fusion process – whether automatic or manual – will allow us to take advantage of the similarity in the underlying problems of data association and combination that span engineering, analysis and cognitive situations.

Recognizing the common elements across the diversity of data fusion problems can provide enormous opportunities for synergistic development. Such synergy – enabling the development of information systems that are cost-effective and trustworthy – requires commonly understood methods for performance evaluation, system engineering methodologies, architecture paradigms, or multi-spectral models of targets and collection systems.

Figure 1: Global State Estimation Problem

\[
\hat{X} = \arg \max \int \lambda (X \mid Z) dX \\
= \arg \max \sum_{k=0}^{\infty} \frac{1}{k!} \lambda \{x_1, \ldots, x_k\} \mid Z dx_1, \ldots, dx_k
\]

Figure 2: Paradigm for Data Fusion and Resource Management Processing Nodes

Figure 3: Characteristic of Many Successful System Designs

3 This prompts the new “DF and RM Dual Node Architecture” name for this architecture.
structure and corresponding nodes. Pattern analysis of the requirements for each node allows selection of appropriate techniques, based on analysis and experience of applicability in the specified conditions.

4 Approaches to building data fusion trees and designing nodes adaptively to mission conditions are discussed in Section 3.3 below.

3. Component Function Design: Design of data fusion nodes, to include specifying data inputs/outputs of component functions (alignment, association and estimation), allocation to human/automatic processes, and technique selection;

Figure 3: Example of System with Integrated Data Fusion and Resource Management Trees

In each phase, analysis of requirements leads to a further functional partitioning. Performance analysis of the resulting point design can lead to further analysis, repartitioning and redesign, or to initiation of the next design phase. Thus, this process is amenable to implementation via waterfall, spiral or other development methods.

The phases of the process, shown in the figure, may be summarized as follows:

1. Operational Architecture Design: System-level problem decomposition; assigning the role for data fusion, as well as for other system functions (sensors, communications, response resources, human operators, etc.);

2. System Architecture Design: Design of the data fusion network (usually a tree) by partitioning the process among C3 nodes and into processing nodes;

4. Detailed Design and Development - Pattern application, algorithm tailoring, software adaptation and development.

It must be admitted that, for all their success in the systematic application of accepted system engineering principles to data fusion system design, the Project Correlation Data Fusion Engineering Guidelines lack formal rigor. The Guidelines provide rules-of-thumb but no rigorous process for generating candidate designs or for evaluating and selecting among them. We hope to establish the basis for such rigor by representing the system engineering process as a type of resource management process.

2 Resource Management

A resource management process is one that combines multiple available actions (e.g. allocation of multiple available resources) over time to maximize some objective function. Such a process must contend with uncertainty in the current situational state and in the predictive consequences of any candidate action. A resource management process will

- develop candidate response plans to respond to estimated world states;

4 Figure 4: Data Fusion Engineering Method (Project Correlation, 1997)
Planning: b) Problem Alignment (Common Referencing) - c) Control: i.e. Plan Execution; generating the control data fusion node: involves functions that directly correspond to those of a
As depicted in Figure 2, a resource management node performs an association/estimation process; a resource management tree performs a planning/execution process. Both these trees – one synthetic (i.e. constructive), the other analytic (i.e. decompositional) are characteristically recursive and hierarchical.

As multi-nodal data fusion trees are useful in partitioning the data association and state estimation problems, so are resource management trees useful in partitioning planning and control problems. A data fusion tree performs an association/estimation process; a resource management tree performs a planning/execution process. Both these trees – one synthetic (i.e. constructive), the other analytic (i.e. decompositional) are characteristically recursive and hierarchical.

As depicted in Figure 2, a resource management node involves functions that directly correspond to those of a data fusion node:
a) Problem Alignment (Common Referencing) - normalizing performance metrics of the given (sub)problem and normalizing performance models of available resources, as well as any control format and spatio/temporal alignment.
b) Planning:
- Plan Generation: Candidate partitioning of the (sub)problem into (sub)subproblems and candidate assignment of resources.
- Plan Evaluation: Evaluating the conditional net cost (e.g. probability of outcome × value of outcome – expected cost of plan execution).
- Plan Selection: determining a decision strategy.
c) Control: i.e. Plan Execution; generating the control commands to implement the selected resource allocation plan.

Planning is a process analogous to Data Association in data fusion. Functions corresponding to Association Hypothesis Generation, Evaluation and Selection are involved: (a) Plan Generation involves searching over a number of possible actions for assembly into candidate plan segments, which are passed to (b) Plan Evaluation and (c) Plan Selection.

As with Hypothesis Generation in data fusion, Plan Generation involves potentially massive searches, which must be constrained in practical systems. The objective is to reduce the number of feasible plans for which a detailed evaluation is required.

Analogous to the selection of fusion trees to constrain the search for association hypotheses, a resource management tree may be selected which searches only over resources, goals and implementation times having high a priori payoff.

The structure of each resource management node is shown in Figure 1. Candidate plans — i.e. schedules of tasking for system resources — are assembled recursively. The level of planning is adapted on the basis of (a) the assessed utility relative to current mission goals of the given plan segment as currently developed and (b) the time available for further planning. By (a), near-term plan segments tend to be constructed in greater detail than far-term ones (for which the expenditure in planning resources may outweigh the confidence that the plan will still be appropriate to the circumstances extant at execution time).

Deeper planning is accomplished by recursively partitioning a goal into candidate sets of sub-goals (Plan Generation) and combining them into a composite higher-level plan (Plan Selection). At each level, candidate plans are evaluated as to their effectiveness in achieving assigned goals, the global value of each respective goal and the cost of implementing each candidate plan (Plan Evaluation). By evaluating these cost/payoff factors to global mission utility, the need for deeper planning or for selection of alternate candidate plans is determined. In many applications, Plan Selection becomes an allocation search function in n-dimensions (i.e. over n-1 future time intervals).

Contentions for assigning available resources are resolved by prioritized rescheduling on the basis of time sensitivity and predicted utility of contending allocations. Each resource management node presents a candidate plan segment for higher-level evaluation. Interlaced data fusion nodes estimate potential side-effects of the plan: e.g. detectable signature changes, flight path changes, or emissions which could interfere with another sensor. Higher-level nodes respond by estimating the impact of such effects on their respective higher-level goals. In this way, plans responsive to global mission goals are assembled in a hierarchical fashion.

In many ways, the least understood part of Problem Decomposition is that of Plan Generation. This is ultimately the problem of systematically finding feasible approaches to given problems. The difficulty derives from
the high-dimensionality of the feasible solution space in challenging real-world planning situations.

Plan Generation ultimately involves novel ways of partitioning problems and applying logical and physical principles in novel ways. Current automated planning systems generate hypotheses via a template method to constrain the Plan evaluation/selection space. This is clearly an area that is ripe for focused investigation.\(^5\)

In comparison with the difficulties of Plan Generation, Plan Evaluation and Plan Selection offer only the challenges similar to those encountered in designing the corresponding data fusion functions. Plan Evaluation challenges are in the derivation of efficient scoring schemes that reflect the expected utility of alternative response plans.

2.2 Information Acquisition Management

We now consider a class of resource management processes most closely linked to data fusion. For lack of a generally accepted term, we may name such processes information acquisition management; by which we include sensor management and data fusion process management (i.e. the JDL model’s level 4).

Following [7-9], information acquisition management can be modeled as a process of choosing a sequence of information acquisition actions; i.e. a strategy \(\alpha\) and a decision function \(\delta: B \rightarrow D\), with a goal to maximize the expected net payoff \(\Omega^*\):

\[
\Omega^*(\alpha, B, \delta) = \sum_{x \in B} \sum_{a \in B} \Phi(x, a, \delta(B), C(B))
\]

where

- \(B = \{B_1, B_2, ... , B_q\}\) An exhaustive partitioning of possible world states \(x \in X; B_i \cap B_j = \emptyset\) for \(i \neq j\);
- \(\Phi: X \rightarrow [0,1]\) Probability density function on possible world states;
- \(D\) The system’s available response decisions;
- \(C: B \rightarrow R\) Cost function for possible utility results \(R\);
- \(\omega^*: X \times D \times R \rightarrow R\) Payoff function.

A cost function \(C\) in an information acquisition system involves allocation of resources, physical risk to resources, information security risks, mutual interference and processor and communications loading and latency, etc.

An information acquisition action \(a \in A\) yields an information set \(Y_a\). A mapping function \(\eta_a: X \rightarrow Y_a\) induces an information structure \(M_a\) on \(X\):

\[
M_a = \{M \mid \exists y[M = \eta_a((\{y\}))]\}.
\]

A sequence of such actions creates a sequence \(B\) of partitions on \(X\), each a refinement on its predecessor:

\[
R(B, \alpha) = \{r \mid \exists M[M \in M_a \& \exists j(B_j \in B \& r = B_j \cap M)]\}.
\]

3 Recasting the Data Fusion System Engineering Problem

The above model for resource management permits a powerful general method for system engineering (e.g. for data fusion system engineering) in terms of a standardized formal representation.

3.1 System Engineering as a Resource Management Problem

Resource management may be defined as a process for determining a mapping from a problem space to a solution space. System engineering is such a process, in which the problem is to build a system to meet a set of requirements. Fundamental to the system engineering process (as in all resource management processes) is a method for representing the structure of a problem in a way that is amenable to a patterned solution.

Among issues in data fusion system engineering are

- Selecting feature sets for exploitation;
- Discovering exploitable context;
- Modeling problem variability;
- Discovering patterns that allow solution generalization;
- Combining uncertain or poorly-modeled information;
- Predicting technique and system performance;
- Predicting development cost and schedule.

System engineering is generally not a discovery process, whereby an idealized essence of the problem is revealed. Rather, system engineering is a process of imposing a candidate structure on a given problem, then evaluating that structure for net utility. Problem structures are generated, evaluated and selected on much the same grounds that scientific theories are: on the basis of

- Conservatism (i.e. conservation of past beliefs or design principles);
- Generality (i.e. breadth of application);
- Simplicity (i.e. Occam’s razor);
• Refutability (i.e. amenability to test).[10]

As a resource management process, system engineering can be implemented as a hierarchical, recursive planning and execution process.

3.2 Data Fusion System Engineering

The Data Fusion (DF) Engineering Guidelines can be thought of as the design specification for a resource management (RM) process; the "phases" being levels in a hierarchical RM tree, as depicted in Figure 5.

A tree-structured RM process is used to build, validate and refine a system concept (which, in our application, may itself be a tree-structured RM or DF process: something of a Universal Turing machine). Each RM node in a system engineering process involves the functions characteristic of all RM nodes:

a) Problem Alignment (Common Referencing) - normalizing performance metrics of the given (sub)problem and normalizing performance models of available resources (e.g. DF tree types or DF techniques – using problem-space/solution-space matrices);

b) Planning: Generating, evaluating and selecting design alternatives (partitioned according to the four design phases shown in Figure 3);

c) Control: i.e. Plan Execution: building or evaluating a DF tree, node or component technique).

The data fusion system engineering process builds, evaluates and selects candidate designs for the system and its components via a hierarchical, recursive process, which permits simultaneous reasoning at multiple levels of depth. The recursive planning process enables both an optimization of design against a given set of requirements and an ability to redesign as requirements change.

The data fusion system engineering process distributes the problem solution into multiple design (i.e. management) nodes. Nodes communicate to accumulate incremental evidence for or against each plausible solution. At any given stage, therefore, the data fusion system engineering process will provide the best plan of action for achieving current goals, consistent with the available situational and procedural knowledge and the available planning and reaction times.

The design process of Figure 4 can be recast as a data structure such as in Figure 5. This will be recognized as a type of fan-out resource management tree, interleaved with a fan-in data fusion tree; similar to that shown in Figure 3. Mission goals and constraints flow leftward from the system level to allocations over successively finer problem partitioning. At each level, a design phase constitutes a grouping of situations (e.g. batches of data) for which responses (design approaches) are coordinated.

This procedure is hierarchical and recursive. Subordinate resource management nodes are activated to develop more detailed candidate design segments when a higher-level node determines that more detailed design is both feasible (in terms of estimated response time and resource cost) and beneficial (in terms of improved likelihood of attaining the assigned goal).

In response to the requirements/constraints flow (right-to-left in the figure), there is a reverse flow, consisting of evaluated candidate design segments; i.e. proposed technique assignments and controls plus estimates of the performance of proposed design segments relative to their assigned goals.

A higher-level node evaluates cost versus benefit (in terms of its own higher-level goal) of the received design proposals and may re-partition its higher goal into a new set of sub-goals if the initial allocations are not achievable. For example, if a required performance against a particular system requirement cannot be met, a new criterion for batching data must be developed.

![Figure 5: Resource Management Tree for Data Fusion System Engineering](image)

The rightward flow of evaluated plan segments is performed by a data fusion tree interleaved with the resource management tree. In the system engineering tree depicted, there can be an evaluation process corresponding to each design node. As in the integrated system shown in Figure 3, planning and control flow leftwards, while estimation and prediction (in this case, design evaluation) flow rightwards.

3.3 Adaptive Data Fusion

An important avenue of research involves the development of adaptive data fusion techniques, by which the selection of data to be processed and of processing techniques to be applied is determined by a system’s resource management process during run-time. In effect, the data fusion tree and nodes are constructed adaptively, based on the system’s assessed current information state and the predicted effectiveness of available techniques to move to a desired information state. Significant work in this area was conducted under the U.S. DARPA Dynamic Multi-User Information Fusion (DMIF) project and
accumulated data refine the support to abstract hypotheses. Thus, abstract hypotheses – the so-called static database – are in fact adaptive to the sensed environment.

The data fusion paradigm permits this adaptivity to be defined in a rigorous way as an organic part of an integrated data fusion. In other words, the model is not an external part of the system. Indeed, the system’s set of models, together with its set of concrete hypotheses at all levels of aggregation combine to form the fusion system’s global hypothesis — its estimation and prediction of the state history of the sensed world.

### 3.4 Coordinated Multi-Level Data Fusion and Resource Management

The above recasting of the system engineering process as a class of resource management processes – coupled with the formal duality between resource management and data fusion – permits a systematic integration of a wide range of engineering and analysis efforts. In effect, this insight allows data fusion techniques to be used in the systematic design and validation of data fusion systems.

It also opens the way for a rigorous approach to the entire discipline of system engineering.

Finally, the search for general methods that span system engineering and in-mission resource management, should lead to a coordinated multi-level approach to real-world problems. The concept is depicted in Figure 7, in the form of the well-known Boyd “OODA”-loop.  

Like a target hypothesis, a target model is ultimately a state estimate, representing the association of a multiplicity of data. Target models are abstract hypotheses; regular target hypotheses are concrete association hypotheses. As accumulated data refines the support and state estimate of concrete hypotheses, so does the assessed impact (i.e. the cost if no change in the current action plan is taken);

- timeliness and other physical constraints; and

6 ‘OODA’ stands for ‘Observe – Orient – Decide – Act’; a paradigm for action that is responsive to a dynamic environment.
resource capability available over time. An integrated resources management process will employ such factors in considering, evaluating and selecting one or more of the action types shown in the figure.

For example, an unanticipated capability in a threat weapon system may be inferred through the sensing and fusion processes. Reaction can include a direct response with weapons, other countermeasures or passive defenses. It will also be desired to refine the characterization of the threat by controlling the data fusion process, coordinated with a modified sensor management plan and corresponding reallocation of communications resources to provide the desired new observation data.

These are all “near real-time” responses, with action plans spanning milliseconds to a few minutes. Modifications in the threat system model may also be needed, even within the course of the tactical engagement. The same new assessment may precipitate longer-term responses: and in the development of system and technology capabilities to address the newly-perceived threat. 7

Analogous to the concept of Global Data Fusion treated in Section 1.1, Global Resource Management will coordinate all the system’s responses at all time scales.

4 Summary – Future Directions

We believe that this paper has opened several new avenues for potential research; including:

- Refinement of the Data Fusion Engineering Guidelines;
- Development of corresponding guidelines for resource management;
- Methods for innovative plan generation
- Developing the formal theory of system engineering;
- Adaptive modeling;
- Adaptive system and technology acquisition;
- Coordinated resource management at multiple levels of granularity;

- Coordinated data fusion at multiple levels of granularity.

We are fortunate to live in a world beset with problems that include a very large number (k) of interesting ones.

References


7 In much the same way as resource management can build coordinated action plans operating at diverse levels of action granularity, so is it possible to perform data fusion that is coordinated at diverse levels of estimation granularity. For example, if a mission objective is to characterize and track a tank column, it may not be necessary to characterize and track each individual vehicle. By associating observations and estimating entities consistently at the appropriate level of granularity, it is possible to reduce combinatorial complexity and, possibly, to recognize features that only emerge at higher levels of aggregation.