

ARMY RESEARCH LABORATORY



Theoretical Considerations for Battlespace Information Mediation

by Richard C. Kaste

ARL-TN-282

July 2007

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory

Aberdeen Proving Ground, MD 21005-5067

ARL-TN-282

July 2007

Theoretical Considerations for Battlespace Information Mediation

Richard C. Kaste

Computational and Information Sciences Directorate, ARL

| REPORT DOCUMENTATION PAGE | | | Form Approved OMB No. 0704-0188 | | |
|---|--------------|-------------------------|---|---|---|
| Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS. | | | | | |
| 1. REPORT DATE (DD-MM-YYYY) July 2007 | | 2. REPORT TYPE Final | | 3. DATES COVERED (From - To) October 2005–May 2006 | |
| 4. TITLE AND SUBTITLE Theoretical Considerations for Battlespace Information Mediation | | | 5a. CONTRACT NUMBER | | |
| | | | 5b. GRANT NUMBER | | |
| | | | 5c. PROGRAM ELEMENT NUMBER | | |
| 6. AUTHOR(S) Richard C. Kaste | | | 5d. PROJECT NUMBER 611102H48 | | |
| | | | 5e. TASK NUMBER | | |
| | | | 5f. WORK UNIT NUMBER | | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: AMSRD-ARL-CI-CT Aberdeen Proving Ground, MD 21005-5067 | | | 8. PERFORMING ORGANIZATION REPORT NUMBER ARL-TN-282 | | |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | | | 10. SPONSOR/MONITOR'S ACRONYM(S) | | |
| | | | 11. SPONSOR/MONITOR'S REPORT NUMBER(S) | | |
| 12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited. | | | | | |
| 13. SUPPLEMENTARY NOTES | | | | | |
| 14. ABSTRACT The purpose of this note is to generate discussions about information fusion modeling for research and hypothesis testing. A modest start on back-of-the-envelope analyses of battlespace digitization and fusion problems is intended to encourage network scientists and analysts of tactical information concerns. We consider several elementary concepts, including amount and value of information, processor level and rate, process rules, and information decay. Given certain theoretical premises about information in a real-time system, the overall problem is one of determining types and connections of processors at various levels to maximize utilizable information. The note utilizes differential equations as a modeling technique; however, other approaches are alluded to. Areas for theoretical investigation are set forth, and several problems are associated with development of information structures and processing techniques. Research merging information theory with control theory may yield opportunities for commercial systems as well as for battle command over the tactical internet. | | | | | |
| 15. SUBJECT TERMS information processing, knowledge fusion, network science, battlespace digitization, cognitive systems | | | | | |
| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | 19b. TELEPHONE NUMBER (Include area code) |
| UNCLASSIFIED | UNCLASSIFIED | UNCLASSIFIED | UL | 38 | Richard C. Kaste 410-278-7781 |

Contents

| | |
|---|-----------|
| List of Figures | iv |
| Acknowledgments | v |
| 1. Introduction | 1 |
| 2. A Paradigm | 3 |
| 3. A Differential Formulation | 4 |
| 4. Types of Processing | 5 |
| 5. Utility and Value of Information | 10 |
| 6. Second-Order Facts | 12 |
| 7. Removing Data From the Buffer | 15 |
| 8. Processing Rates | 16 |
| 9. Subtasks, Stress, and Degradation | 17 |
| 10. Accuracy | 21 |
| 11. Efficiency and Compounding | 22 |
| 12. A Final Example | 28 |
| 13. Closing Thoughts | 30 |
| Bibliography | 33 |
| Distribution List | 35 |

List of Figures

| | |
|--|----|
| Figure 1. An abstract information-processing system. | 4 |
| Figure 2. A more sophisticated information-processing network..... | 6 |
| Figure 3. Two notional configurations..... | 7 |
| Figure 4. A functional processing flow..... | 7 |
| Figure 5. A processor sequence. | 29 |
| Figure 6. Shunting of low-order facts..... | 30 |

Acknowledgments

The author would like to acknowledge the contributions of John Brand, George Hartwig, Eric Heilman, and Mark Thomas. Conversations with these researchers helped shape some of the developments. Mr. Hartwig, in particular, offered valuable suggestions that led to reformulating this note.

INTENTIONALLY LEFT BLANK.

1. Introduction

All aspects of information are increasingly vital to U.S. force employment, with measures and countermeasures involving digital information becoming as important as actual weapons systems. Battlespace digitization involves rapid transfer of data among sensors, intelligence officers, commanders, and weapons. Our Southwest Asian conflicts show that modern warfare requires improved knowledge processing. Vast quantities of disparate information can be utilized via intelligent systems. Information is routinely computerized syntactically and semantically. More sophisticated processing systems could include formal reasoning and associative connections.

We envisage this note as one of a series of notes leading to a unified approach for building a model of battlespace information mediation, a term which connotes conveyance via intermediary mechanisms and includes notions of filtering, summarization, fusion, and inference. We intend to develop approaches involving the following:

- division of decision-making responsibility in complex real-time processors
- self-organizing cognitive software systems encoding knowledge
- propagation of information through inference nets
- selective querying internal to the system based on perceived utility to the reasoning being performed
- consideration of cognitive constructs as vector-algebraic objects to be manipulated symbolically (including derivation of measures of divergence from expectations and hence detection of deception)
- calculations of the values of weapons and tactics (indeed, of information itself) during a conflict based on actual battlespace parameters

The purpose of this note is to generate discussion on information fusion modeling for research and hypothesis testing. As such, it does not represent our final thinking. Certain formulations are, perhaps, too simplistic; but this is just an initial approach to studies that could utilize realistic data in battle command testbeds. The work is intended to encourage network scientists of the Directorate, researchers qualified as analysts of tactical information concerns. Network theory (especially as applied to information fusion) appears to be in a relatively primitive state of development, and scaling what solutions do exist into postulated battlespace requirements is poorly understood. This work is a modest start on back-of-the-envelope analyses of battlespace digitization and knowledge fusion problems.

We consider several elementary concepts and extensions, including the following:

- amount and value of information
- processor level and rate
- process rules (e.g., separation, consolidation)
- information decay
- process tasks, comprising subtasks that may interact
- completion time and accuracy
- time-independent degradation factors that affect subtasks
- time-dependent stress function
- state characteristic
- efficiency

It is apparent that such investigations lend themselves to leveraging design techniques for computer operating systems and search engines; consider such methods in the context of what follows.

Several assumptions tend to arise based on tactical considerations. For instance, most data will be generated at the lower levels and most data will not result in information at the higher levels. Similarly, information processing should generally be pushed down to the lowest level possible, in turn minimizing higher-level overload. Moreover, in analyzing the ability to get data to where it is needed, bandwidth considerations are important. We hope that the processing model developed from theoretical constructs we are examining can be used to address such assumptions and yield at least qualitatively meaningful insights.

Given certain theoretical premises about information in a real-time system, the overall problem is one of determining types and connections of processors at various levels to maximize utilizable information. Related problems involve minimizing time through the net and minimizing process cost. In considering using low-level data by high-level processors and production and utility of second-order facts, we are interested in developing results analytically where possible. Another aspect of this is conceptual: for instance, second-order facts may connote disambiguation of sensor input on the one hand or reasoned knowledge on the other. This note utilizes differential equations as a modeling technique; however, other methodological approaches will be alluded to.

2. A Paradigm

We are interested in representing development of metainformation based on an accumulation of information within a network of communicating nodes. Observations could be related in many ways: no relation, sibling, parent, child, includes, included-in, equals, same-level, etc. To illustrate some characteristics of an information network, consider the following situation. Processors (devices for filtering, summarization, fusion, and inference) receive raw (possibly-related) data from the environment. Each accumulates a database and processes it into first-order facts (possibly at a rate proportional to that of data reception) with the object of producing second-order facts for a higher processor level. Queries and responses take place among processors. Information utility may decay with time, and there may be information loss if processors or networks are overloaded. As first-order facts are formed, gaps are realized in the formation of second-order facts. Queries are made to other processors, possibly at a rate proportional to the first-order formation rate. As second-order facts are completed, the processor moves them to the next level.

Less abstractly, we might consider that to know the battlespace is to know the characteristics of all units at all times. Each unit comprises subunits (down to some smallest) having various properties such as position, velocity, strength, and attachment as functions of time. Sensors pick up characteristics of the smallest subunits, and processors combine the information. Information itself has properties such as veracity, timeliness, and applicability. Processor mechanisms include simplification (e.g., filtering), consolidation (e.g., averaging sensings for a unit position), and separation (e.g., observations by a single sensor may yield information for distinct processors).

One way of conceptualizing processor types involves the real-world notion of echelonment. The lowest-level processor might be associated with brigade combat team (BCT) information and be interested only in the recent past as it reflects fast-moving local battles. Higher-level processors might be associated with higher echelons and, in developing an overall battlespace picture, would accumulate an historical database.

Processing increases pragmatic content, and information value depends on level of application. Moreover, information quality may be related to amounts obtained from different sources. Second-order development calculi must account for interactions among information amount, utility, and content. For instance, not all facts generated heuristically are useful or even meaningful. Processing can certainly yield additional facts; whether it is able to increase the amount of information existing in the raw data depends on one's semantics. A higher-order fact will generally be of greater utility at its level than are the component data, even though in an information-theoretic sense there may be less information.

An important and difficult problem, then, is: given time-varying observations of the smallest subunits, what is the optimal configuration of processing mechanisms? We might choose between the realms of data mining (pattern-recognition, statistical summarizations) on the one hand, and of knowledge discovery (inference without preconceived notions, development of interesting aspects of the raw data) on the other. Without concerning ourselves with such (somewhat overlapping) distinctions at this point, we proceed with developing an abstract paradigm.

3. A Differential Formulation

We consider the situation in figure 1:

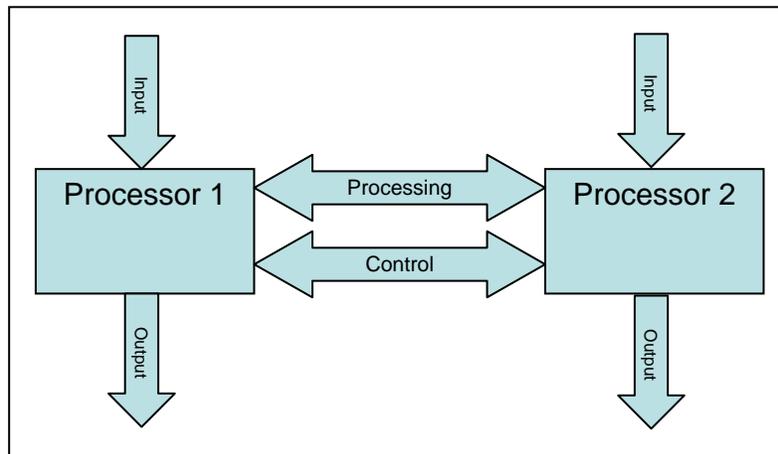


Figure 1. An abstract information-processing system.

Let I_i = information that node i has obtained, raw input

P_i = information node i is processing, partial data

O_i = output facts from node, i fully formed conclusions

Q_i = queries from node i

A_i = answers from node i

$\frac{dI_i}{dt}$ = forcing function (of battlespace, sensor)

Note all these are functions of time. Moreover, $\frac{dP_i}{dt}$ is a function of P_i and Q_j , Q_i , and O_i are functions of P_i and A_j , and A_i is a function of P_i , O_i , and Q_j .

Assume (initially) there is no redundant information and that all information is correct. We are interested in the movement of information from input, through the processing, to output: $O_i(t)$ is

desired. Moreover, we are concerned only with continuous amounts of information; this formulation does not consider that discrete information is buffered in reality.

As a start at modeling a processor, we consider that $\frac{dP_1}{dt} = f(I_1, A_2, P_1)$. That is, we assume the output rate is a function of the amount being processed and we ignore (for the time being) the impact of outside queries (as opposed to answers) on the processing. To limit the rate we would probably like $\frac{dO_1}{dt}$ to increase monotonically and asymptotically as P_1 increases. In relating queries and answers we note, for a simple linear formulation, that Q_i may be a fraction of P_i and be based on I_i and A_j , and that A_i may be a fraction of P_i and be based on Q_j . Pending resolution of whether query rate can be considered, to a first approximation, a linear function of the amount being processed we can now write a differential formulation of such a system as:

$$\left\{ P_i = I_i + A_j + Q_j - O_i, \frac{dO_i}{dt} = P_{\max} \left(I - e^{-P^R} \right), \frac{dQ_i}{dt} = q(I_i + A_j), \frac{dA_i}{dt} = aP_j \frac{dQ_i}{dt} \right\}. \quad (1)$$

Of course, the situation leading to this formulation is simplistic. The main purpose of this note is to sketch concepts for possibly enhancing the development of more realistic models of information mediation. As a transition into such discussions, figure 2 shows a schematic that is more representative of the real world.

4. Types of Processing

Why use parallel or distributed processing, as opposed to one large processor? Some intuitive responses present themselves. Bandwidth constraints will generally preclude the latter approach. Access time for desired information tends to increase with system size (given that the information is in the local database). The probability of obtaining desired information tends to increase with time and with system size. A large system may eliminate redundant information more readily. Since processors, in some sense, determine relationships among inputs we seek to compare the advantages of notional configurations in figure 3.

The nature of our processing is that information is extracted from real-time input and combined with data from a database of recent extractions. Rules must be developed for database formation and combination. We distinguish a processor using only a near-real-time buffer from one having access to an accumulated database.

Assume information takes the form of n units (undefined data points, or information-theoretical bits) of utilizable data per m observations. When these observations are processed, in a sense they emerge stretched or compressed. Information loss may occur if the processor is overloaded.

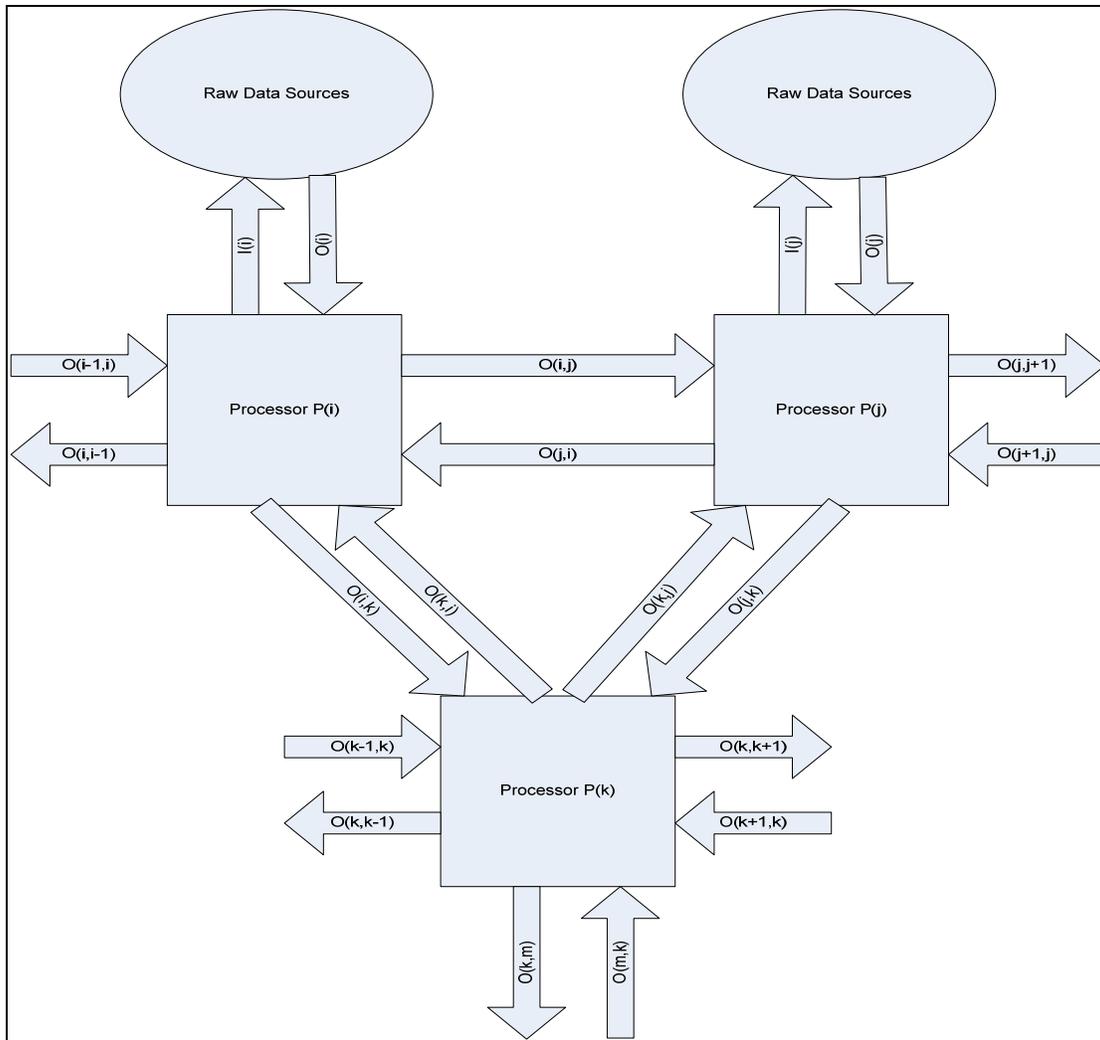


Figure 2. A more sophisticated information-processing network.

A main derived result is reduction in rawness, that is, increase in utility. The next level has available n^* units per m^* observations. Outputs may have different utilities when applied to different processors; some notion of utility matrix must be developed.

Processing may involve simplifying information from one source (e.g., smoothing data as with a filter) or consolidating information from several sources (e.g., averaging positions of targets). Processing may also involve separating types of information (e.g., observations by a single sensor may be split into information about two units for shipment to separate higher-level processors). The point is that a processor may be thought of as a function. For instance, we consider flows like in figure 4, in which Roman letters represent functions, Greek letters represent fractions, and + and * represent operators.

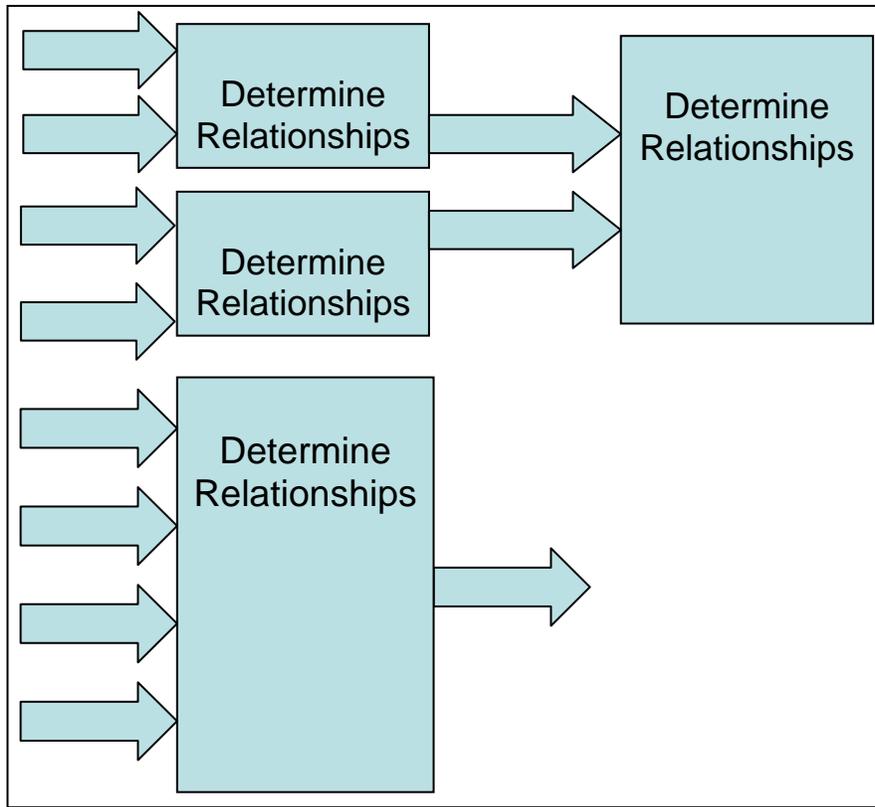


Figure 3. Two notional configurations.

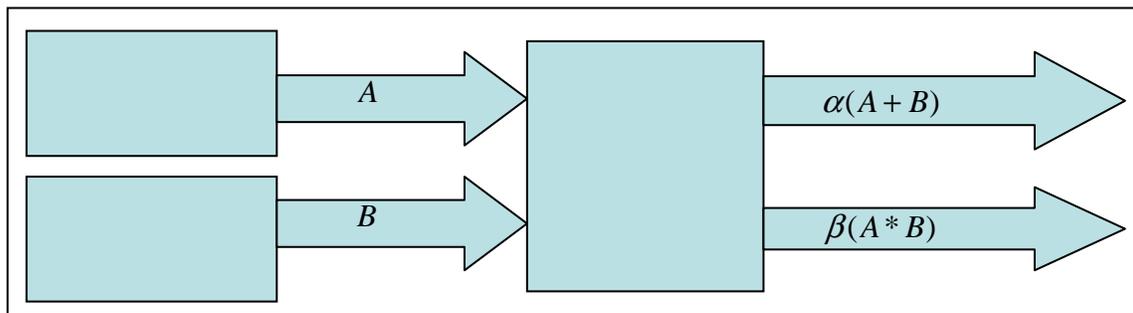


Figure 4. A functional processing flow.

Speculating further on the development of first-order and second-order facts, we might develop

$$\binom{m}{n} \equiv \frac{m!}{n!(m-n)!}$$

derivative facts from m raw data points, given that n facts are required to

generate one derivative. The rate of first-order fact development depends on input rate and amount being processed, e.g., it may be directly proportional to input rate and increase to an asymptote with the amount being processed. If n first-order facts are needed to yield one second-order fact but some are unavailable, then querying other processors might fill in the gaps

at a rate proportional to the amount of second-order facts. However, we must account for overlap.

Note also that staleness can be accounted for in terms of processor load. Suppose the rate in is I and the rate out is F . If $I \equiv F$, staleness should equal 0. If $I < F$, then in some sense staleness should also equal 0 (the system is waiting for enough input to yield output). We could, therefore, say $I < F \Rightarrow F \equiv I$. If $I > F$, we have build-up in the buffer going as $\int_0^t (I - F) dt$.

We can initially define the amount of information in the database as the time integral of rate of raw information input. For example, if the input rate is a constant k , we have $D(t) = \int_0^t k dt = kt$. By formation of facts, raw information might be cleared out, for shipment to the next processor level or for storage in a higher-level-fact database.

Utility accumulates similarly, but we must account for decay with time. We can develop (via discrete constant input) expressions such as

$$D(t) = \int_0^t k_a dt - \int_0^t \int_s^t k_d dt ds \quad (2)$$

and

$$D(t) = \int_0^t k_a dt - \int_0^t \int_s^t k_d (1 - e^{-t}) dt ds, \quad (3)$$

based on constant and exponential decay, respectively. In general, we may write

$$D(t) = \int_0^t f(t) dt - \int_0^t \int_s^t d(t) dt ds \quad (4)$$

for utility in the base.

As a processor at a fixed level develops facts, partial information is formed. Some will be filled by new input, some will be filled by answers from querying other processors, and some will be unfillable. Of the completed and partial information in a processor, some will be information that other processors require, so we must model the waiting time for completion of a partial fact.

Since the processing network comprises a connected set of nodes, we can represent flow from node i to node j by a 1 ij -entry in a square matrix of zeros and ones. With this representation, the n th power of the matrix yields entries with the total number of n -stage paths from node i to j . Further, we can consider real or fractional entries as representing the amount or fraction of information shipped. Some representation of hierarchy can be produced if for each pair $i \neq j$, we have $a_{ij} = 1$ if $a_{ji} = 0$ —there need be no transitivity. A matrix of transmission capabilities/capacities may be necessary if there is to be treatment of redistributing inputs to avoid low throughput.

A variety of functional forms could represent the output rate of a processor versus input rate; e.g., linear increase, monotonic increase to asymptote, increase then decrease. This last form

may be particularly useful as it reflects the intuitive notion that a certain threshold amount of data is required for generating higher-order facts but that too much clogs the processor.

A differential formulation simpler than the one developed previously involves

$$\left\{ \frac{dF}{dt} = \alpha \frac{dI}{dt} - \beta F, \frac{dS}{dt} = \gamma \frac{dF}{dt} - \delta S \right\}, \quad (5)$$

where F and S are first- and second-order facts, $\frac{dI}{dt}$ is given input rate, and there is no consideration of querying or degradation.

Both these equations have the form

$$f' = \alpha g - \beta f, \quad (6)$$

with solution

$$f = \frac{\alpha \int g e^{\beta t} dt + c}{e^{\beta t}}. \quad (7)$$

Suppose $g(t)$ is a constant k . Then,

$$f = \frac{\alpha k \int e^{\beta t} dt + c}{e^{\beta t}} = \frac{\alpha k}{\beta} + \frac{c}{e^{\beta t}}. \quad (8)$$

Assuming $f = 0$ at $t = 0$ yields

$$f = \frac{\alpha k (e^{\beta t} - 1)}{\beta e^{\beta t}}. \quad (9)$$

Since $f \rightarrow \frac{\alpha k}{\beta}$ as $t \rightarrow \infty$, we have an example of self-limiting behavior. Given $g \equiv k$, we would

like the processing to stabilize or max out with raw data being lost (or accumulated in the database for subsequent processing with decayed utility).

We need to represent the fact that when raw data are combined, they are removed (or databased).

Suppose input arrives at a rate $\frac{dI}{dt} = k$, is combined, and leaves at a rate αm , where m is the

amount combined. Clearly, buildup occurs if $\alpha m < k$. The rate of change of amount of information in the processor is

$$\frac{dP}{dt} = \frac{dI}{dt} - \alpha m. \quad (10)$$

(In saying that m raw facts are combined at some rate to yield one higher-order fact we assume that the m are available.) Thus,

$$P(t) = I(t) - \alpha mt + c, \quad (11)$$

and, if $\frac{dI}{dt} = k$ and $P(0) = 0$, then

$$P(t) = (k - \alpha m)t. \quad (12)$$

We may represent staleness or decay of information in the processor by the amount of time information spends in the buffer.

We have avoided mention of the content of facts being processed, and from an abstract information theoretical point of view, this simplifies matters. However, for complex systems, we must consider heterogeneous data (e.g., separate processing of sensings of vehicles and observations of individuals). Considering context is more difficult yet. Extracted or derived information is a more optimal coding (in an information theoretic sense), and it contains only part of the source information. For example, given two positions, extracted data could be exemplified by the distance between them.

In maximizing utility output, we must consider that information becomes stale if it moves slowly through the system. We want an expression for the decay of utility at a fixed level of processing. Intuitively, if we set utility to 1 at time 0, then at times close to 0, utility should be close to 1. Utility should decay to near 0 in finite time and approach 0 asymptotically. Another complication arises when considering that the decay rate of a high-order fact may differ from those of corresponding low-order facts. Information decay is a difficult theoretical problem based partly on the nature of the process generating the facts. Decay involving utility of a low-order fact by a low- or high-order processor may be more straightforward. We must also consider that permanent facts (forming a fundamental basis for the processing, as opposed to ephemeral or temporarily-useful facts) may be thought of as having zero decay. Moreover, decay as related to utility may be nonmonotonic, even highly so in certain tactical situations.

5. Utility and Value of Information

Several important ideas involve the concept of information value. We will attempt to use theoretical concepts of information rate and information content to develop this concept. Intuitively, pragmatic content of (battlespace) information will generally depend on time, and processing may increase pragmatic information. Information value depends on the level of application. Moreover, utility or quality of information will generally be related to staleness and to the amounts of information coming from different sources.

One conventional definition of information is simply a numerical measure of the uncertainty of an experimental outcome. This suffices for our purposes—we can say that information has value

to the extent it is useful in changing an outcome. Of course, this is a difficult proposition that precipitates considerations of potential use and of measuring or predicting outcome change.

We must clarify the basic terms information content and utility and construct a mathematical model by which we can speak of information being produced and transmitted. We receive information when informed of an event whose occurrence was uncertain. Utility carries with it this notion, as well as one of applicability, which generally refers to appropriateness of the data to a given processor level (e.g., knowledge that a certain BCT is under attack may not be important at Corps). Usefulness generally involves closeness to the present situation (e.g., that a vehicle is now at some position may be of low usefulness in 2 days).

Information can be thought of as just a tool, an item of neutral value without regard to the circumstances to which it is being applied. Another way to consider information value is analogously to the value of a test—the difference between the expected gain of a process if the test is conducted and the expected gain if no test is run.

Such economic metaphors are pervasive. We can analyze information overload in terms of marginal cost vs. marginal utility. Another economic approach is to consider that information is worth what people are willing to pay for it—it has no inherent value except in some context. A trivial example is that knowledge of a state capital may be worth a million dollars on a game show but could have no value in a military situation. Indeed, in the military situation, it could conceivably have negative value, say, in the sense of distraction. One avenue of investigation involves the notion of value being expressed as the product of utility (of a fact, like as a tool in achieving a purpose) and benefit (of that purpose, like in accomplishing a mission), with both of these attributes subject to experimental evaluation.

A matrix formulation can be postulated in terms of contexts C_i and pieces of information I_j —the contextual value is V_{ij} . (We are at this point purposefully vague about units of information and facts. We intend to tie these notions into mechanisms involving symbolic propagation of potential utility or numerical propagation analogously to Bayesian nets.) As an example, let us consider the value of gasoline. If we need to operate a gasoline engine, it may have great value. If we need to extinguish a fire, it may have, at most, zero value. Similarly, water has little value for the engine and some value for the fire context. So it is for information.

One way to think of utility is to presume that information has value only with respect to complete knowledge. Suppose total knowledge of a sector is represented by one observation from each of n subsectors. We might consider parallel presentation as yielding the linear utility $u = (\text{subsectors reporting})/n$, but this does not reflect synergism or decay. Another approach is to compute expected utility as the product of utility and probability of the application. We can consider the contribution of various pieces of information to the value of the outcome. Assuming truth (droppable in more realistic analyses) and that redundant information contributes

nothing, compressive processing should yield, when compared to input, a lower rate of observations each of higher absolute utility.

Let the utility of a raw data point be unity at the lowest level. Then, if m_i nonoverlapping $(i-1)$ -level observations are required to generate one fact at level i , we could take as the value of a raw point for level 2 m_2^{-1} , for level 3 $(m_2 m_3)^{-1}$, etc. Of course, this notion holds only as an average, attributable to justifiable concerns over the number of higher-level facts that may actually result from a single raw data point.

Information-theoretic considerations may help in conceptualizing change of utility as a result of changing levels or of processing. A unary experiment contains $\log_2(2 \text{ outcomes}) = 1$ bit of information, and an experiment with n outcomes yields $\log_2 n$ bits. Although processing can compress information or yield additional facts, whether it can increase the amount of preexisting information depends on the frame of reference. We are interested in questions like ‘when a first-level processor yields one second-level fact from m first-order facts, what happens to the absolute utility?’ and ‘can it be said that a system trades off absolute utility for speed?’. We want to reflect the idea that some facts are more useful than others. Also, if we use explicit character sets, a decrease/increase in the size of the set requires longer/shorter sequences to contain the same information.

Suppose node i supplies information at a rate $F_i(t)$ to some higher-level node. Then the total input rate at that node is ΣF_i . Now, if the rates change to F_i^* the relative quality could be expressed as $\frac{\Sigma F_i^*}{\Sigma F_i}$. It seems plausible that several detailed observations would have greater

utility to a higher level together after processing than they would as separate observations. Utility, at least in the battlespace, may be considered a decreasing function of (increasing) level and time. We need a method for representing composite utility of several facts (at several stalenesses). Note that utility per se is independent of amount.

6. Second-Order Facts

Given a (constant, initially) database, how does a processor develop second-order facts? Intuitively, if decay is excluded, all possible facts should be developed (asymptotically) in a manner depending on the processor and amount of information. The amount of data has the positive attribute of making more information available for completion and, if large, the negative attribute of tending to overload the system.

We examine some methods of combination, but note that one approach is simply to assume a function with certain reasonable properties. For example, more facts should arise from more

information, and no second-order facts arise from one raw fact. It is probably reasonable to consider that output rate is a function only of the amount in the processor.

Suppose there are n tanks in a unit. One observed tank might be considered to yield $1/n$ amount of information about the unit. Given two observations of tanks, several situations are possible: the tanks are different, yielding $2/n$ information, or the tanks are the same, yielding $1/n$.

Extending this for m observations, a weighted average arises, considering that a random sample with replacement has probability of no repetition

$$\frac{n(n-1)\dots(n-m+1)}{n^m}. \quad (13)$$

Thus, for two observations, we have the probability of separate tanks as

$$\frac{n(n-1)}{n^2} = \frac{n-1}{n} \quad (14)$$

and of the same tank $\frac{1}{n}$, so the information could be thought of as having value

$$\frac{n-1}{n} \cdot \frac{2}{n} + \frac{1}{n} \cdot \frac{1}{n} = \frac{2n-1}{n^2}. \quad (15)$$

With regard to missing information, suppose m facts are needed for a higher-level fact. Then for $n \geq m$ facts available, $\frac{n!}{m!(n-m)!}$ higher-level facts are inherent. Toward bounding the

conceptual model, we might think in terms of $\frac{n!}{[m-k]! [n-(m-k)]!}$ potential facts with $k \leq m$

observations missing. Considerations such as these may yield thoughts about changes in knowledge brought about by adding observations when a certain number have already been received, leading to proper queries to other processors. If missing information is related to a sector, it may be inversely proportional to the amount received. We must be cautious about dealing with the probability of a requested fact filling in knowledge gaps; the content of the information, not just the amount, may have to be considered.

We need theoretical justification for combining units in this manner. One approach is simply to consider that one fact is associated with each subset of a database: the number of second-order facts associated with n information points is 2^n . With this formulation, one fact (absence of information) is associated with the empty set.

However facts are developed, we should consider basic words (undefined terms), basic sentences (axioms), and logical rules in examining the ways sequences of situations develop in time. We should be able to represent that phenomena comprise others. Any second-order development calculus must account for interactions among amount, utility, and content. For instance, not all

facts generated by some simple rule of thumb will be useful or even meaningful. Perhaps function fitting to input/output of real-world processors would assist in developing the theory.

As mentioned, multiple observations might be more easily stripped out of the system at higher levels. Another sort of redundancy involves observation or querying for information derivable from existing facts. An essential part of developing an optimal network is understanding how lower-level facts might be processed directly at a higher level. Such facts would increase the amount in a higher-level buffer similarly to the contribution of regular facts, but may be processed less efficiently—if a processor is not intended to process relatively raw information, that information has lower relative utility. It could be argued that if a processor must perform low-level functions to obtain its regular data, this would diminish other processing. On the other hand, when low-level facts are considered at this level, a different processing methodology might exist to use them directly. In any event, database development and querying would take different forms. Possibly the same amount of data could be processed regardless of its intended level, but low-level input would have less utility at a higher level. We will have to consider using high-order data by low-order processors, particularly in a network that permits inter-level querying.

Consider the generated utility of a processor-buffer. Assume that data come into the buffer with two attributes: amount (as time-rate of input) and utility (also a function of time, but independent of amount). Assume the value of a generated fact at the next level is simply a multiple of the average utility in the buffer at the time of generation. Thus, if we do not remove facts or consider utility decay, we can write average utility as

$$A(t) = \frac{\int_0^t r(t)u(t) dt}{\int_0^t r(t) dt}, \quad (16)$$

where the buffer is initially empty, $r(t)$ is the input rate, and $u(t)$ is utility.

Note that using average utility in generating higher-order facts in effect assumes instantaneous access. It may be argued that the actual buffer should comprise data only within some time-window. We may be able to leverage methods for analysis of computer algorithms to incorporate more realistic access as part of processing cost. There should be additional examination of the presumption that a set of n facts of $1/n$ utility each is in some sense equivalent to one fact of unity utility. We desire consistency between utility-decay of separate and consolidated facts. We must distinguish among utility as a function of level, utility at a fixed level as a function of time, and information content. For instance, can we say that n facts of average utility a equal one higher-order fact of utility $\frac{an}{m}$, where m data points comprise one higher-order fact? We need to reflect that a higher-order fact may be of greater utility at its level than the component data points, even though in a sense there may be less information.

7. Removing Data From the Buffer

For a dynamic battlespace, facts would be removed from the buffer as the situation changed and their utilities diminished. Removal after utility reaches some non-zero minimum may be considered, but we first consider removal after some time t_m following entry into the buffer. We can write

$$\frac{dB}{dt}(t) = r(t) - r(t + t_m) \quad (17)$$

as the rate of change of amount of data in the buffer (with no decay). Thus, for

$$0 \leq t \leq t_m, \quad (18)$$

$$B(t) = \int_0^t r(t) dt. \quad (19)$$

Afterwards,

$$B(t) = \int_{t_m}^t r(t) dt - \int_{t_m}^t r(t + t_m) dt \quad (20)$$

or

$$B(t) = \int_{t_m}^t r(t) dt - \int_0^{t-t_m} r(t) dt. \quad (21)$$

Thus, we have

$$A(t) = \frac{\int_{t_m}^t r(t)u(t) dt - \int_0^{t-t_m} r(t)u(t) dt}{\int_{t_m}^t r(t) dt - \int_0^{t-t_m} r(t) dt} \quad (22)$$

as the average utility of facts in the buffer at $t \geq t_m$. This (appropriately multiplied) could be used as $u(t)$ for the next level.

Many mechanisms exist for clearing buffers. For instance, with regard to context, it may be that for a small group of related facts simply no more inferences can be drawn. Also with regard to overload problems, we may want to place facts into a database or shunt them to another processor, possibly at a different level.

Looking now at the buffer in terms of the time a unit of information spends there, we see that given a beginning amount $B(t_b) = B_b$, input rate $r_i(t)$, and output rate $r_o(t)$, the amount in the buffer at some ending time is $B_b + \int_{t_b}^t [r_i(s) - r_o(s)] ds$. Therefore, the average time spent in the buffer is

$$\frac{\int_{t_b}^{t_e} \left\{ B_b + \int_{t_b}^t [r_i(s) - r_o(s)] ds \right\} t dt}{\int_{t_b}^{t_e} \left\{ B_b + \int_{t_b}^t [r_i(s) - r_o(s)] ds \right\} dt}. \quad (23)$$

This may then be converted to utility. Of course, r_i and r_o may be modeled as functions of parameters other than time: e.g., $r_o(B)$. Such dependencies yield more interwoven formulations of average time and utility. We may be more interested in distributions of times or utilities at the expense of complicating the representation.

Consider utility decay with time, initially without removal. A differential amount of data $r(0)dt$ enters at $t = 0$ with utility $u(0)$. The information then decays with rate $v(u(0), t)$, and utility of the differential amount after time x could be written $u(0) - \int_0^x v(u(0), s) ds$. Similarly, for data entering the buffer at any time t the utility at time x is $u(t) - \int_t^x v(u(t), s) ds$. Therefore, one measure of the product of amount and utility in the buffer at time x is $\int_0^x r(t) \int_t^x v(u(t), s) ds dt$. Moreover, we could reasonably remove data from the buffer based on actual utility rather than time. This must be pursued for realistic analyses.

8. Processing Rates

Consider the time a unit of information takes to traverse a dual-level processor. The unit arrives at the first buffer, with capacity a , at time t_o and departs at $t_o + \frac{a}{m}$, which we consider also as instantaneous arrival at the second buffer, with capacity b . The unit leaves the b buffer similarly. We wish to examine $a(t)$ and $b(t)$, assuming $a(t_o) = a_o$ and $b(t_o) = b_o$.

In subsequent evaluations, we consider compressed facts vs. original facts with regard to information, keeping in mind that processing rate is independent of the number f of first-order facts comprising a second-order fact. When the unit arrives at b , we have

$$b = b_o - n \left(t_o + \frac{a_o}{m} \right) + \frac{m}{f} \left(t_o + \frac{a_o}{m} \right), \quad (24)$$

the original amount minus output during time of interest plus input during time of interest. The time of departure of the unit from b reduces to

$$\left(t_o + \frac{a_o}{m} \right) \frac{m}{nf} + \frac{b_o}{n}. \quad (25)$$

Denoting this by t^* and letting $t_o = 0$, we have

$$t^* = \frac{b_o f + a_o}{fn}. \quad (26)$$

Note this time is independent of the a -processing rate. Further, t^* is inversely proportional to the b -processing rate, and writing

$$t^* = \frac{b_o}{n} + \frac{a_o}{fn} \quad (27)$$

shows it to be roughly inversely proportional to the comprisal number f . Thus, the throughput time of a unit in this situation can be reduced by increasing n or f .

Now consider decay of first-order information moving through a constant-rate processor. With “decay parameter” α , we may write the value of a data unit as $u(t) = e^{-\alpha(t-t_o)}$. When this datum

clears the a -processor at time $t_o + \frac{a_o}{m}$, it has value, $e^{-\frac{\alpha a_o}{m}}$, assuming $t_o = 0$. This can be a

prototype example of utility, staleness, and amount of information while passing through a series of buffers/processors.

Assume a processor has some maximum buffer size. When the input and processing rates would cause it to be exceeded, one has several model options. One is sloughing input: resetting to a value (e.g., 0) that allows processing or to some maximum, either absolute or situation-dependent. Another is redirection to another processor at the same level. Another, more sophisticated, solution is to remove buffered facts before their normal expiration. Finally, the incoming data could be shunted to a higher-level processor.

It might be argued that output (at least a higher-level processor) should be time-shifted, with a corresponding decay in utility, to reflect non-real-time processing. Possibly another way to represent lag is to consider that the buffer used in generating higher-order facts involves data arriving only within some time window.

9. Subtasks, Stress, and Degradation

Assume an information-processing task is a series of n nonoverlapping simple subtasks, where the i th subtask normally takes time τ to complete. Imposing some degradation, initially time-independent, define the degradation factor d_i as the ratio of degraded completion time τ'_i to normal completion time. Then, the degraded completion time of the task is $\sum_{i=1}^n d_i \tau_i$. Further, let degradation be a constant d_j for a fraction s_j of a subtask (where $\sum s_j = 1$), and let τ denote undegraded completion time. Then degraded $\tau' = \tau \sum_j d_j s_j$.

Note that the strenuousness of a task might be reflected by a stress function. For example, a light task could be represented by $y = k_l t$ (k_l large), a moderately-difficult task could be represented

by $y = k_m t$ (k_m medium) or $y = k_m (1 - e^{-m_m t})$ (m_m large), and a strenuous task could be represented by $y = k_s (1 - e^{-m_s t})$ (m_s small). An overall stress function probably exhibits time-dependence. Intuitively, efficiency could be described by expressions like $f(t) = 1 - kt$, $f(t) = k(t+1)^{-1}$, and $f(t) = e^{-kt}$. (Section 12 of this note considers the notion of efficiency in a somewhat different manner.) The reciprocal of efficiency could be considered a (time) degradation function: $d(t) \equiv [f(x)]^{-1}$. In general, if efficiency is $f(t)$ for $t \in [t_1, t_2]$, we might have $\tau' = \tau \int_{t_1}^{t_2} [f(x)]^{-1} dx$.

Consider a task in terms of the amount $y(t)$ of processing accomplished over time. Consider an efficiency function $f(t)$ portraying stress on the processor in terms of relative ability to perform the task. One might then propose this model for the degraded processing function:

$$y'(t) = \int_{t_0}^{t_0+t} \dot{y}(t) f(t) dt . \quad (28)$$

The following are two examples:

$$y = kt, f = e^{-mt}, t_0 = 0 \Rightarrow y' = \frac{k}{m} (1 - e^{-mt}) \quad (29)$$

and

$$y = kt, f = 1 - mt, (\because t = 1/m \Rightarrow f = 0), t_0 = 0 \Rightarrow y' = \frac{kt}{2} (2 - mt) . \quad (30)$$

We now we have an expression for the processing accomplished under ordinary and degraded conditions. We would like a transform such that, given ordinary time t to complete y units of processing, we can compute degraded time t' to accomplish y' units. We try

$$y' = g(t) \Rightarrow t' = g^{-1}(y'), \quad (31)$$

where g^{-1} denotes the inverse function. Then, the first example yields

$$t' = -\frac{1}{m} \ln \left(1 - \frac{my'}{k} \right), \quad (32)$$

and the second yields

$$t' = \frac{1 - \left(1 - \frac{2my'}{k} \right)^{\frac{1}{2}}}{m}. \quad (33)$$

Consider a subtask in terms of the fraction $s(t)$ accomplished. In an unstressed environment we might have, normalizing for $t \in [0, \tau]$, $s(t) = \tau^{-1}t$. Assume a stressing effect

$$\frac{ds(t)}{dt} = f(t), \quad (34)$$

that is,

$$s(t) = \int_0^t f(x) dx. \quad (35)$$

Again, in an unstressed environment

$$\frac{ds(t)}{dt} = \tau^{-1} \quad (36)$$

and

$$s(t) = \tau^{-1}t \quad (37)$$

as t goes from 0 to τ .

Accomplishing a fraction $s(t^*)$ that would ordinarily be accomplished by time t^* will take with degradation time $\int_0^{t^*} d(x)dx$. Clearly, for efficiency $f(t) \equiv 1$ the degraded time is t^* . Under unstressed conditions, we have

$$\frac{\int_0^{t^*} dx}{\int_0^{\tau} dx} = \frac{t^*}{\tau}. \quad (38)$$

Under stressed conditions, the fraction accomplished by t^* can be expressed as

$$\frac{\int_0^{t^*} f(x) dx}{\int_0^{\tau} f(x) dx}. \quad (39)$$

We can now answer the question ‘how much will be accomplished in an interval of length t^* ?’ by

$$s'(t^*) = \frac{\int_{t_0}^{t_0+t^*} f(x) dx}{\int_{t_0}^{t_0+\tau} f(x) dx}, \quad (40)$$

where t_0 is clock time of subtask initiation. The question ‘how long will it take to accomplish a fraction $s'(t^*)$ of the task?’ involves the more complicated expression

$$F(t_0 + t^*) = s'(t^*)F(t_0 + \tau) + [1 - s'(t^*)]F(t_0), \quad (41)$$

where $F(x) \equiv \int f(x)dx$. The (generally non-closed-form) solution for t^* depends on $f(t)$.

Given efficiency as a function of time and time for undegraded completion, what is the time for completion with overall degradation? That is, what is the transformation of the subtask

summation $\sum_{i=1}^n d_i \tau_i$? One might say, letting t_i denote clock time of the initiation of the i th subtask,

$$\sum_{i=1}^n \left[d_i \tau_i \int_{t_i}^{t_i+1} d(x) dx \right]. \quad (42)$$

But how is t_{i+1} determined? Given that transformed $\tau'_1 \equiv \tau_1$ is it true that

$$\tau'_{i+1} = \tau'_i d_i \tau_i \int_{t_i}^{t_i+1} d(x) dx? \quad (43)$$

The time to accomplish a given fraction of a subtask is stretched by the overall stress function in such a way that (letting $t_{i+1} = t_i + \Delta t$):

$$\begin{aligned} \Delta t'_1 &= d(0) \Delta t, & t'_1 &= t_0 + \Delta t'_1; \\ \Delta t'_2 &= d(t'_1) \Delta t, & t'_2 &= t_1 + \Delta t'_2; \\ & & &= (t_0 + \Delta t'_1) + d(t_0 + \Delta t'_1) \Delta t \\ & & &= [t_0 + d(0) \Delta t] + d[t_0 + d(0) \Delta t] \Delta t; \end{aligned} \quad (44)$$

and so forth. We need a better mapping of the fraction of subtask and task accomplished to account for this time modification.

Suppose we have the task of performing n subtasks simultaneously. What is a reasonable model of task degradation d , given d_i for each subtask i ? If the subtasks are independent, perhaps

$\max_i \{d_i\}$ would do; if they interact, we try a different approach.

Consider a matrix $A_{n \times n} \equiv [a_{ij}]$, where a_{ij} represents the effect that subtask i has on subtask j : if i is degraded one unit then j is degraded a_{ij} units. (We postpone briefly a discussion of what is meant by degraded one unit.) Let us now impose degradation in the form of an n -dimensional vector $\bar{x} \equiv [x_1, x_2 \dots x_n]$ and write the interacting degradation as $\bar{x} A$. This would seem to indicate that, for example, subtask 1 is degraded x_1 units due to subtask 1 (since, clearly, A 's diagonal consists of ones), plus $a_{21} x_2$ units due to subtask 2, plus $a_{31} x_3$ units by task 3, and so on. Then, for the overall task, perhaps the weighted sum $\bar{w}(\bar{x} A)^T$ might represent the overall degradation, where $\bar{w} \equiv [w_1, w_2 \dots w_n]$ and $\sum_{i=1}^n w_i = 1$, depending on importance of subtasks.

Let us now return to degraded one unit. We have discussed the degradation factor $d \equiv \tau' / \tau$. Consider this example:

$$[1, 2] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [1, 2], \quad (45)$$

with $w_1 = 0.2$ and $w_2 = 0.8$. Here, we have overall degradation $0.2 + 1.6 = 1.8$. Given that subtask 2, degraded, takes twice as long as normal, is it reasonable that the overall task takes $1.8 \times$ as long? Apparently, overall degradation must mean something different in this simultaneous formulation.

10. Accuracy

Intuitively, one might expect accuracy to decrease with time and task difficulty. Can we account for the effects of decreased accuracy due to degradation? Perhaps inaccuracy in one subtask affects all subsequent subtasks. Inaccuracy in a component subtask probably affects the whole task. Inaccuracy (perhaps defined as measured deviation from normal at a given fraction of subtask completion) could be treated analogously to time dilation. Increased error frequency is probably a discretization of inaccuracy.

Another way to think of accuracy is in terms of a time-varying Poisson distribution. The invariant Poisson distribution

$$p(x) = \frac{\lambda^x e^{-\lambda}}{x!} \quad (46)$$

gives the probability x errors will occur in some time interval, under the following assumptions. Consider a fixed unit of time T in which errors may occur. We assume errors occur independently and that for a short Δt , the probability of one error is proportionate to the length of the interval, i.e., equals $\frac{k}{T} \Delta t$, with k constant during T . We also assume the probability of two or more errors during Δt is negligible. Then it can be shown that the earlier distribution is obtained, where $\lambda = \frac{k}{T} t$, and t is the interval length. Here, $\frac{k}{T}$ is the number of errors per unit time, as derived empirically in an unstressed environment over the period T . For a time-varying situation, $\frac{k}{T}$ and λ are functions of time.

Suppose

$$\frac{k}{T} = c\tau, \quad (47)$$

τ measuring clock time from the beginning of stress. A formal substitution yields

$$p(x) = \frac{ct}{x!} \tau^x (e^{-c\tau})^\tau. \quad (48)$$

However, we are really interested in an interval $[\tau_0, \tau_0 + t]$ for which λ is a function of τ . Can we use for $\frac{k}{T}$ an average of $\frac{k}{T}$ over the interval? Formal substitution into the time-invariant equation of the expression $\frac{1}{t} \int_{\tau_0}^{\tau_0+t} h(\tau) d\tau$, where $h(\tau)$ is an error-rate function, yields

$$p(x) = \frac{H^x e^{-H}}{x!}. \quad (49)$$

Here is another instance in which plotting specific distributions, along with time-invariant $\frac{k}{T}$, may yield additional insights.

11. Efficiency and Compounding

Let us now write the efficiency of task performance as

$$E = \frac{w}{w + R}, \quad (50)$$

where w is time spent actually processing and R is idle time. It is apparent that, in general, the overall efficiency of a task comprising a (nonoverlapping) series of subtasks is not the sum of the subefficiencies total

$$E_T \equiv \frac{\sum w_i}{\sum w_i + \sum R_i} \neq \sum \frac{w_i}{w_i + R_i}. \quad (51)$$

Although two sets of efficiencies generated by different times can yield the same total efficiency

$$\left(\text{e.g., } \frac{n}{n+n} = \frac{m}{m+m}\right), \quad (52)$$

this is not generally true. Consider that

$$R = \frac{w(1-E)}{E}; \quad (53)$$

now

$$\frac{w_1 + w_2}{(w_1 + w_2) + \left[\frac{w_1(1-E)}{E} + R_2 \right]} = \frac{w_1 + w_2}{(w_1/E) + w_2 + R_2}, \quad (54)$$

a function of w_1 . That is, total efficiency depends on specific subtask times.

Suppose we write degraded efficiency

$$E_d = \frac{D_w w}{D_w w + D_R R}, \quad (55)$$

where degradation factors D_w and D_R are greater than or equal to unity. It turns out that

$$E_d = \frac{D_w}{D_w + D_R \left(\frac{1-E}{E} \right)}. \quad (56)$$

However, the fact that $R = 0$ when $E = 1$ flaws this expression for E_d , since $D_R R = 0$ when $E = 1$.

Suppose we try

$$E_d \equiv \frac{D_w w}{D_w w + (R + R_D)}, \quad (57)$$

where R_D is additional idle time in the degraded mode. Now, this cannot lead to an expression purely in terms of E ; the best we can do is something like

$$E_d = \frac{T D_w}{w(D_w - 1) + T + R_D}, \quad (58)$$

where $T \equiv w + R$, or

$$E_d = \frac{D_w E T}{T[D_w E + (1-E)] + R_D}. \quad (59)$$

Some simplification results by letting $T \equiv 1$ then,

$$E_d = \frac{D_w E}{D_w E + (1-E) + R_D}. \quad (60)$$

However, this formulation also suffers, as exemplified by

$$E = \frac{.5}{.5 + .5} = .5, \quad (61)$$

and

$$E_d = \frac{(1.5)(.5)}{(1.5)(.5) + (.5 + .25)} = .5 \quad (62)$$

(increasing idle time and processing time yields the same efficiency). Previously, we had

$$E_d \equiv \frac{D_w w}{D_w w + (R + R_D)}. \quad (63)$$

In an unstressed situation, $E_d = E$. For $D_w = 1$, we have

$$E_d = \frac{w}{w + (R + R_D)}; \quad (64)$$

doubling idle time yields

$$\frac{E_d}{E} = \frac{w + R}{w + 2R}, \quad (65)$$

a seemingly acceptable ratio. So, with this formulation,

$$E_d = \frac{ET}{T[D_w E + (1 - E)] + R_D}. \quad (66)$$

Let us return to the original expression

$$E_d = \frac{D_w}{D_w + D_R \left(\frac{1 - E}{E} \right)}, \quad (67)$$

where $R > 0$. In an unstressed situation, $E_d \equiv E$. For $D_w = 1$, we have

$$E_d = \frac{w}{w + D_R R}; \quad (68)$$

doubling idle time again yields the seemingly acceptable ratio

$$\frac{E_d}{E} = \frac{w + R}{w + 2R}. \quad (69)$$

Let us, as before, take the numerator of the first E_d expression to be w ; we can then derive

$$E_d = \frac{1}{D_w + D_R \left(\frac{1 - E}{E} \right)}, \quad (70)$$

where $E \neq 0$ and $R \neq 0 \Rightarrow E \neq 1$. With this formulation, $E_d \rightarrow 0$ as $E \rightarrow 0$ and

$E_d \rightarrow \frac{1}{D_w}$ as $E \rightarrow 1$. Plots of the two expressions for E_d based on several values of D_w , D_R , and R_D would be a useful illustration.

Let us continue with our analysis of the expression

$$E_d = \frac{w}{D_w w + D_R R}. \quad (71)$$

We have from

$$E_d = \frac{1}{D_w + D_R \left(\frac{1 - E}{E} \right)} \quad (72)$$

that

$$\frac{dE_d}{dE} = \frac{D_R}{[ED_w + D_R(1-E)]^2}. \quad (73)$$

Since this is always greater than 0, E_d increases monotonically with E . Also, it can be shown that the second derivative is always positive— E_d as a function of E is concave upward, and the whole curve lies above the line $E_d = \frac{E}{D_R}$. This model implies that degrading lower efficiency is always worse than degrading higher efficiency—tasks which are performed well to begin with are more resilient.

What does

$$E_d = \frac{w}{D_w w + D_R R} \quad (74)$$

mean with regard to completion time? What does multiplying E_d by a constant mean operationally? We have from

$$T_D \equiv D_w w + D_R R = \frac{w}{E_d} \quad (75)$$

that completion time for what is normally accomplished in w (undegraded processing time) is inversely proportional to E_d . Also, since

$$T = \frac{w}{E} \quad (76)$$

and

$$T_D = \frac{w}{E_d}, \quad (77)$$

we have

$$T_D = \frac{E_T}{E_D}, \quad (78)$$

so that apparently undegraded efficiency is necessary to compute degraded completion time.

Ordinarily for a subtask, we have

$$E_i = \frac{w_i}{w_i + R_i}. \quad (79)$$

Suppose we consider subtasks performed simultaneously until the total task is accomplished. Suppose attention is divided among subtasks according to factors such that $\sum \alpha_i \equiv 1$. Then w_i

might be transformed to $\frac{w_i}{\alpha_i}$. With regard to idle time, we might assume a linear relationship between (transformed) w_i and R_i until proven otherwise. But in this context, what is meant by combined idle time? Perhaps $\max_i \{R_i\}$ would be a reasonable formulation.

Perhaps an analog to the electrical resistance law

$$R_T = \frac{1}{\sum \frac{1}{R_i}} \quad (80)$$

could be used for compounding efficiencies, since in a sense, the reciprocal of efficiency is a measure of resistance to task completion. The formulations

$$\frac{1}{E_T} = \frac{1}{\sum \left(\frac{1}{E_i} \right)} \quad (81)$$

and

$$E_T = \frac{1}{\sum \left(\frac{1}{E_i} \right)} \quad (82)$$

are flawed, since total efficiency greater than unity is possible. Trying

$$E_T = \frac{1}{\sum \frac{1}{E_i}} \quad (83)$$

yields

$$E_T = \frac{E_1 E_2}{E_1 + E_2} \quad (84)$$

for the two-subtask case, and for n subtasks of equal E , we have

$$E_T = \frac{E}{n}, \quad (85)$$

so this analog may offer possibilities. Of course, it may be argued that our basic concept of compounded efficiency is too simplistic, if only because it does not consider nonindependence of subtasks.

Considering that 0 resistance connotes $E = 1$ and infinite resistance $E = 0$, we might try resistance in accordance with

$$\frac{1}{E} - 1 \equiv \frac{1-E}{E}. \quad (86)$$

Looking at

$$E_T = \frac{1}{\sum \frac{1}{E_i}} \quad (87)$$

again, we would have

$$\frac{1-E_T}{E_T} = \frac{1}{\sum \frac{E_i}{1-E_i}}; \quad (88)$$

this works for the single subtask, but cannot be used with $E_i = 1$. For n subtasks of equal E , we have

$$\frac{1-E_T}{E_T} = \frac{1}{n \frac{E}{1-E}} \Rightarrow E_T = \frac{nE}{1+(n-1)E}. \quad (89)$$

It is apparent this yields increasing values of E_T with increasing n . Let us try

$$E_T = \frac{E}{1+(n-1)E}, \quad (90)$$

where we justify the division by n as an n -way splitting of attention.

We have from

$$\frac{1-E_T}{E_T} = \frac{1}{\sum \frac{E_i}{1-E_i}} \quad (91)$$

that

$$E_T = \frac{\sum \frac{E_i}{1-E_i}}{1 + \sum \frac{E_i}{1-E_i}}. \quad (92)$$

For equal attention factors we may divide this by n ; but, we need to account for nonequal attention factors, so consider the expression

$$E_T = \frac{\sum \frac{E_i/n}{1-E_i/n}}{1 + \sum \frac{E_i/n}{1-E_i/n}} \equiv \frac{\sum \frac{E_i}{n-E_i}}{1 + \sum \frac{E_i}{n-E_i}}, \quad (93)$$

which for equal efficiencies simplifies to $\frac{nE}{n+(n-1)E}$.

Generalizing gives

$$E_T = \frac{\sum \frac{\alpha_i E_i}{1-\alpha_i E_i}}{1 + \sum \frac{\alpha_i E_i}{1-\alpha_i E_i}}. \quad (94)$$

However, this formulation still suffers from a need to compensate for values of E_T greater than E_i . Also, note all this derivation is without considering degradation. Investigations into analyzing subtask adaptation and compounding are essential to further modeling.

12. A Final Example

As an exercise in setting up a throughput analysis (one that helps shape the conclusion), consider the processing situation in figure 5, where the first processor is characterized by

$$\begin{cases} g'(t) = pa^\mu(t) \\ a'(t) = f'(t) - mg'(t) \end{cases} \quad (95)$$

and the second by

$$\begin{cases} h'(t) = qb^v(t) \\ b'(t) = g'(t) - nh'(t) \end{cases}. \quad (96)$$

We have

$$a = f - mg, \quad (97)$$

so

$$g' = p(f - mg)^\mu. \quad (98)$$

The simplifications of assuming a constant driver

$$f'(t) = k, \quad (99)$$

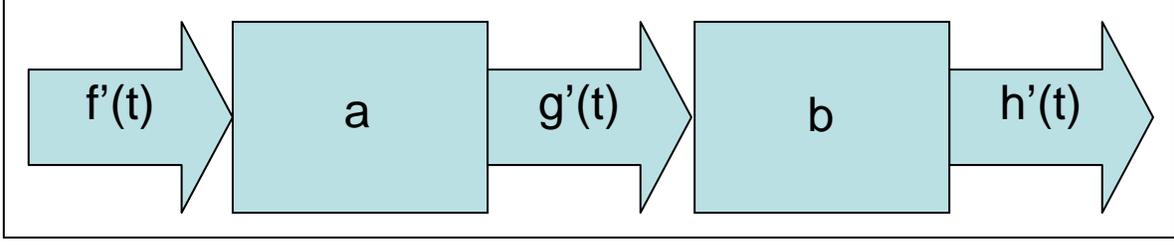


Figure 5. A processor sequence.

$$\mu \equiv 1, \quad (100)$$

and

$$g(0) = 0 \quad (101)$$

yield

$$g = \frac{k}{m^2 p} [(mpt - 1) + e^{mpt}]. \quad (102)$$

Continuing, we have

$$b = g - nh, \quad (103)$$

and assuming $\nu \equiv 1$ gives

$$h' = q(g - nh). \quad (104)$$

With the just-obtained solution for g we can solve for

$$h = e^{-nqt} \left[\frac{\alpha q e^{nqt - \beta t}}{nq - \beta} + \frac{\alpha \beta (nqt - 1) e^{nqt}}{n^2 q} - \frac{\alpha e^{nqt}}{n} + c \right], \quad (105)$$

where $\alpha = \frac{k}{m^2 p}$ and $\beta = mp$. Proceeding in this manner, a (cumbersome) expression for $h(t)$ can be derived. Visualization techniques will certainly have applications in analyses of semi-realistic networks, and simulation is seen to be useful, if not essential, as network complexity increases.

As a processing function that involves more self-limiting behavior, reconsider the situation

$$a'(t) = f' - \alpha(1 - e^{-a}). \quad (106)$$

Then, for constant $f' \equiv k$, we have

$$\frac{da}{dt} = (k - \alpha) + \alpha e^{-a}, \quad (107)$$

which implies

$$a = \ln \left(\frac{\exp \left[(k-a)(t+c) - \alpha \right]}{k-\alpha} \right). \quad (108)$$

For $a(0) = 0$, we find

$$c = \frac{\ln(k-\alpha) + \alpha}{k-\alpha}. \quad (109)$$

A modification of this problem that would build on the discussion of use of low-order facts by high-order processors can be illustrated by figure 6.

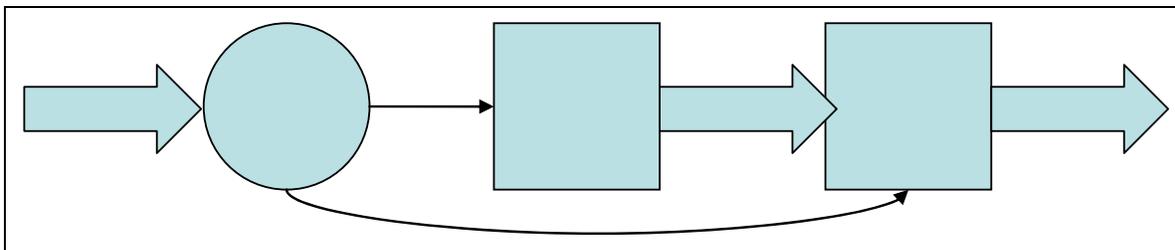


Figure 6. Shunting of low-order facts.

The circle indicates a decision concerning shunting of low-order facts to the separate processor levels. The decision must be optimized within some constraints (e.g., binary switch, fractional separation as a function of input rate). A rewording of part of our overall problem is: what parameter values of qualitative relationships make one level preferable to another?

13. Closing Thoughts

Obviously, this work is mostly abstract development, but the intent is to allow general application to diverse data. Areas for theoretical investigation include the following:

- calculus of variations
- catastrophe theory
- cellular automata
- coding theory
- control theory
- cybernetics

- decision theory
- fractals
- fuzzy sets
- game theory
- information theory
- mathematical programming (in particular dynamic programming)
- measure theory
- networks (their topology and geometry)
- queuing theory
- scheduling
- stochastic processes
- systems theory

We intend to synergize with U.S. Army Research Laboratory (ARL) efforts on computerized simulation of data fusion networks and self-organizing sensor communications.

Many opportunities exist for theoretical and applied work. Is it possible, under certain situations, to develop a transform of rate in to utility out? The basic problem of time-tagging information for decay as it moves through the system is challenging. For instance, if it takes a certain amount to generate output, what is the time a unit of information spends in the processor? For query consideration, multiple input/output ports should be modeled. Expression of degradation via differential equations may alleviate certain difficulties. If task accomplishment $y(t)$ is the desired solution, we could consider the amount, rather than fraction, of task completed. We are also interested in processor rate as an input and in solving for $\frac{dy}{dt}$ to yield maximum

accomplishment or minimum time. It is probable t may not be the only independent variable, e.g., some environmental factor θ may enter as a parameter in the stress function, perhaps

$$g(t) = k_1 e^{k_2 \theta} f(t). \quad (110)$$

We are also interested in the state of the processor performing a task, e.g., in order to remove the processor from action before damage can occur. Although probably associated with overall degradation, we might want state characteristic to be a function of processing rate—one formulation is

$$h(t) = k_1 (1 + k_2 e^{k_3 t}) \frac{ds}{dt}. \quad (111)$$

Also, accuracy and completion time are probably associated with state characteristic.

An eventual goal is a self-organizing cognitive software system encoding knowledge as multidimensional threads and maximizing information storage and access via automatic transformations of interim associations. With a system of graph-theoretical nodes (objects) and arcs (relationships), information can be recalled through association—when one representation is activated, even with a fragmented pattern, so are others with common nodes. The effort entails topological, semantic, and set-theoretical aspects. The knowledge structure, dynamically reconfigurable, would process facts, rules, and relations among database information with query-resolution algorithms. Bidirectional traversal of arcs could be based on attributes, quantification, negation, context, synonyms/commonalities, supersets, and applicable processes.

Models of knowledge processing are needed for intelligent control. It is not clear how to obtain a network minimizing nodal distance, separate knowledge bases for information-routing and the application per se, account heuristically for expected queries, suspend actual computation as the system reconfigures structures, or resolve centralized/distributed control trade-offs. Information decay is an interesting theoretical problem (e.g., decay rate of a high-order fact may differ from that of corresponding low-order facts).

Several problems are associated with developing information structures and techniques for efficient query processing. Perhaps rules approximating local answers can minimize data transferred among nodes. Future research can explore structuring techniques less restrictive than subnetting and include (tactically important) vague knowledge. Indeed, the Advanced Decision Architectures Collaborative Technology Alliance (CTA) evaluation panel recommended increasing research into computational models of conceptualizations. Investigations into detecting deception, measures of deviation from a plan, and text retrieval all indicate a fruitful area of exploration involves representing information as vectors in a space of basis concepts.

Continued research is intended to complement ongoing experiments and modeling efforts. The work benefits information science and technology in areas like fusion, incremental databasing, and parallel query resolution. It supports the ARL mission via fundamental research to provide the U.S. Army necessary analytical support. In particular, it addresses our information technology goals of analysis and assimilation to help reduce the commander's uncertainty. It furthers ARL's strategic plan by focusing on key areas of digitization and communications science and by its intent to utilize the larger CTA organization and results. This research toward merging information theory with control theory may yield opportunities for upgraded or new commercial systems as well as for battle command over the tactical internet.

Bibliography

- Aguirregabiria, J. Dynamics Solver. <http://tp.lc.ehu.es/jma/ds/ds.html> (accessed 21 June 2007).
- Alberts, D.; Garstka, J.; Stein, F. *Network Centric Warfare: Developing and Leveraging Information Superiority, 2nd edition*; C4ISR Cooperative Research Program: Washington, DC, 1999.
- Boisot, M. *Information Space*; Routledge: New York, NY, 1995.
- Boisot, M. *Knowledge Assets*; Oxford University Press: New York, NY, 1998.
- Boltyanskii, V. *Optimal Control of Discrete Systems*; Keter: Jerusalem, 1978.
- Borrelli, R.; Coleman, C. *Differential Equations: A Modeling Perspective*; Wiley: New York, NY, 1996.
- Brand, J. *A Proposed Modeling Protocol for Evaluating Information Attacks*; ARL-TR-112; U.S. Army Research Laboratory: Aberdeen Proving Ground, MD, 1999.
- Brodeen, A.; Frank, D. *Target Prioritization to Optimize Expected Utility for a Simple Battle Scenario*; BRL-MR-3945; U.S. Army Ballistics Research Laboratory: Aberdeen Proving Ground, MD, 1991.
- Brogan, W. *Modern Control Theory*. Quantum: New York, NY, 1974
- Committee on Network Science for Future Army Applications, National Research Council. *Network Science*; National Academy Press: Washington, DC, 2006.
- Findeisen, W.; Bailey, F.; Brdys, M.; Malinowski, K.; Tatjewski, P.; Wozniak, A. *Control and Coordination in Hierarchical Systems*. Wiley: New York, NY, 1980.
- Hall, D.; McMullen, S. *Mathematical Techniques in Multisensor Data Fusion, 2nd edition*; Artech House: Boston, MA, 2004.
- Headquarters, Department of the Army. *Mathematics of Military Action, Operations, and Systems*; DA Pamphlet no. 70-5; Arlington, VA, 1968.
- Hestenes, M. *Calculus of Variations and Optimal Control Theory*. Wiley: New York, NY, 1966.
- Ivanishchev, V.; Krasnoshchekov, A. *Control of Variable Structure Networks*. Optimization Software: New York, NY, 1987.
- Johnson, S.; Levis, A., Eds. *Science of Command and Control: Coping with Uncertainty*; AFCEA International Press: Washington, DC, 1988.

- Kaste, R.; Hansen, C. *A Model-Based Approach to Battle Execution Monitoring*; ARL-TR-3394; U.S. Army Research Laboratory: Aberdeen Proving Ground, MD, 2005.
- Kocak, H. *Differential and Difference Equations through Computer Experiments, 2nd edition*; Springer-Verlag: New York, NY, 1989.
- Nemytskii, V.; Stepanov, V. *Qualitative Theory of Differential Equations*; Dover: New York, NY, 1989.
- OASD-NII, Command and Control Research Program Website. <http://www.dodccrp.org> (accessed 21 June 2007).
- Press, W.; Flannery, B.; Teukolsky, S.; Vetterling, W. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press: New York, NY, 1986.
- Scherl, R.; Ulery, D. *Technologies for Army Knowledge Fusion*; ARL-TR-3279; U.S. Army Research Laboratory: Aberdeen Proving Ground, MD, 2004.
- Shannon, C. A Mathematical Theory of Communication. *Bell System Technical Journal* **1948**, 27, 379–423, 623–656.
- Thompson, J. *Empirical Model Building*. Wiley: New York, NY, 1989.
- Turner, R. *Logics for Artificial Intelligence*. Wiley: New York, NY, 1985.
- Walsh, D. Fusion for the Army. PowerPoint Presentation v. 2004.2, MITRE, 2004.
- Wylie, C.; Barrett, L. *Advanced Engineering Mathematics, 6th edition*; McGraw-Hill: New York, NY, 1995.

NO. OF
COPIES ORGANIZATION

1 DEFENSE TECHNICAL
(PDF INFORMATION CTR
ONLY) DTIC OCA
8725 JOHN J KINGMAN RD
STE 0944
FORT BELVOIR VA 22060-6218

1 US ARMY RSRCH DEV &
ENGRG CMD
SYSTEMS OF SYSTEMS
INTEGRATION
AMSRD SS T
6000 6TH ST STE 100
FORT BELVOIR VA 22060-5608

1 DIRECTOR
US ARMY RESEARCH LAB
IMNE ALC IMS
2800 POWDER MILL RD
ADELPHI MD 20783-1197

3 DIRECTOR
US ARMY RESEARCH LAB
AMSRD ARL CI OK TL
2800 POWDER MILL RD
ADELPHI MD 20783-1197

ABERDEEN PROVING GROUND

1 DIR USARL
AMSRD ARL CI OK TP (BLDG 4600)

NO. OF
COPIES ORGANIZATION

1 DIRECTOR
USARL
AMSRD ARL CI CN
A CLARK
2800 POWDER MILL RD
ADELPHI MD 20783-1197

1 DIRECTOR
USARL
AMSRD ARL CI
J GOWENS
2800 POWDER MILL RD
ADELPHI MD 20783-1197

1 DAPE-MR
B KNAPP
300 ARMY PENTAGON
RM 2C489
WASHINGTON DC 20310

1 US ARO
AMSRD ARL RO MM
J LAVERY
DURHAM NC 27703

1 USARDECOM
CERDEC C2D
I MAYK
FT MONMOUTH NJ 07703

1 USARDECOM
CERDEC C2D
T NEWSOME
FT MONMOUTH NJ 07703

1 USARDECOM
CERDEC C2D
J SOOS
FT MONMOUTH NJ 07703

1 USARL
AMSRD ARL HR-S
M STRUB
ALEXANDRIA VA

1 DIRECTOR
USARL
AMSRD CI CB
L TOKARCIK
2800 POWDER MILL RD
ADELPHI MD 20783-1197

NO. OF
COPIES ORGANIZATION

ABERDEEN PROVING GROUND

20 DIR USARL
AMSRD ARL CI C
B BROOME
AMSRD ARL CI CS
D ULERY
AMSRD ARL CI CT
B BODT
J BRAND
T HANRATTY
C HANSEN
E HEILMAN
R KASTE (5 CPS)
M MITTRICK
J O'MAY
J RICHARDSON
M THOMAS
AMSRD ARL HR
P DEITZ
AMSRD ARL HR M
E BOWMAN
AMSRD ARL HR S
L PIERCE
AMSRD ARL WM BA
A THOMPSON