APPLYING INTENT-SENSITIVE POLICY TO AUTOMATED RESOURCE ALLOCATION: COMMAND, COMMUNICATION AND MOST IMPORTANTLY, CONTROL

Harry B. Funk
SMA Information Flow Technologies (SIFT)
2119 Oliver Ave. S.
Minneapolis, MN 55405-2440
hfunk@SIFTech.com

Dr. Christopher A. Miller
Dr. James P. Richardson
Dr. Christopher L. Johnson
John Shackleton
Honeywell Technology Center
3660 Technology Drive
Minneapolis, MN 55418
{cmiller,jrichardson,cjohnson,jshackleton}@htc.honeywell.com

ABSTRACT

As battlefield communications technologies have begun to achieve their potential, the notion of 'winning the information war' has begun to take on a new complexion—one in which we may be our own biggest enemy. A fundamental problem is that the overhead costs and complexity associated with deciding how to allocate information resources—including generation, processing and routing resources—are too high in an environment where human decision and guidance resources are already overstressed.

We are developing an approach that enables commanders to specify policy that will inform an automated communications resource management system. This policy represents the commander's intent for the allocation of communications resources during the execution of a mission; the policy takes the form of a set of general and specific statements about the priorities, constraints and objectives for information flow. We view this policy as central to any problem in which a decision-maker wishes to precisely guide the behavior of a resource allocation actor. As such, policy has utility in complex domains ranging from cockpit display space to refinery operations. Our approach is called IPSO-FACTO—Intuitive Policy Specification for Optimized Flow of Asynchronous C^3I Transmissions in Operations[1]*. IPSO-FACTO addresses problems including: suitable representations of policy, policy conformance metrics, adaptive information allocation, multi-user policies, and semi-automated policy construction.*

We have implemented a prototype version of the IPSO-FACTO system (that does not yet incorporate this task-based policy derivation) and are currently evaluating it. Initial results show that the policy representation we have provided is very expressive, but time consuming and error prone to generate directly. Nevertheless, policy created in such a fashion does enable human control of automated resource allocation algorithms and can improve the performance of such resource allocation dramatically.

INTRODUCTION

As sensor and processor capabilities continue to increase, the amount of current, potentially relevant information continues to exceed the available communications bandwidth. Even when a user is **dedicated** to the task of deciding what information is worthy of transmission (e.g., when surfing the web), the analysis task is simply too big, and the results are often not quite what was intended. In an environment where human decision and guidance resources are already overstressed, the overhead costs and complexity associated with deciding how to allocate information resources are simply unaffordable.

While this problem exists for most actors on the battlefield, it is most critical for the tactical commander. The commander could delegate this task to subordinates or automation, but this invites mismatches between the commander's goals and intentions and the information policies that are enforced. Intelligent systems are being developed to better manage communication networks, but how can a commander successfully convey his intentions to one of these complex systems, so that the crucial information gets to the right soldier at the right time?

In our view, the solution must comprise the following components:

- **Policy Representation**—a syntactic formulation that balances the need for expressivity and comprehensibility,
- **Policy Conformance Metrics**—a computational framework that allows evaluation of a given proposed solution against the expressed policy,
- **Adaptive Information Control (AIC)**—a resource allocation mechanism that is sensitive to the interaction of expressed policy with world state,
- **Multi-user Policies** —a method to allow multiple users with differing interests and scopes of control authority to work collaboratively to establish a single effective policy,

---

[1] This work was performed under funding from DARPA contract DABT63-99C-0003

- **Semi-automated Policy Elaboration and Conversion (SPEC)**—a means to assist a user in specifying what may be a fairly detailed policy in a rapid, intuitive fashion.

DESIGN

Adaptive Information Control (AIC)—IPSO-FACTO is a part of DARPA's larger Agile Information Control Environment (AICE) program. By providing importance values in keeping with commander's intent, IPSO-FACTO allows other elements of the AICE system to be adaptive to changing battle plans, conditions or commander policy-revising the allocation of communications resources just as the destruction or jamming of a transmission station can. In order for this allocation to follow the commander's intent, the interpretation of policy by the allocation mechanism must, of course, match the semantics the commander used to formulate the policy. The AICE program solved this issue by creating an encapsulated policy formulation that shielded the AIC module from needing comprehensive battle state information.

Policy Representation— In today's operations, policy guidance is conveyed from the commander to subordinates by means of something like an OPORD, a structured free text explanation of the planned operation and the associated implications. Unfortunately, as is demonstrated again and again in operations, this free text conveyance is often misstated and/or misinterpreted (to a greater or lesser degree). This occurs even though the humans involved share common training and context information. For future systems which pair a human with automation, this common basis is reduced, and the potential for misinterpretation grows.

What is needed is a more precise, mathematical formulation of policy, one in which the syntax and semantics are well defined and unambiguous. We have created such a formulation, based on the terms and operators that we felt a commander would commonly use to discuss his intent concerning the use of communication resources.

This representation is conceptually illustrated in Figure 1. Each commander's policy is created as a set of statements (individual 'policy elements') each of which assigns an importance (or value) function to a defined sub-region in a multidimensional space.

Regions may be based on a single dimension ('Requests for weather information [Content] get Importance 0.2') or on a combination of dimensions ('Requests owned by the Zone Reconnaissance task [Owner] for weather information [Content] from Satellite 476B [Source] to 3rd Air Calvary Division [Destination] get Importance 0.8). If the policy element regions are allowed to overlap, then they must be sequenced (typically from most to least specific) to indicate the order of precedence.

More formally, an information request r_k is of the form $(w_k, s_k, \mathbf{d}_k, c_k, u_k)$, where:

	Owner	**Source**	**Destination**	**Content**	**Importance**	
(	{W1}	{S1}	{D1}	{C1}	.9	)
(	{W2}	{S2}	{D2}	{C2}	.8	)
(	{W3}	{S3}	{D3}	{C3}	.5	)
(	*	*	*	*	.1	)

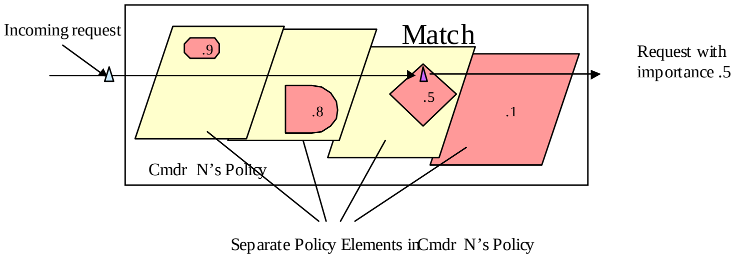


Figure 1. Representation of policy in IPSO-FACTO.

- w_k is the *owner*—the creator of the information request. The set of all possible owners is W.
- s_k is the *source*—an application entity that participates in an information transmission. The set of all possible sources is the set of applications A.
- $\mathbf{d}_k$ is the *set of destinations*—a set of application entities that participate in an information transmission. The set of all possible destinations is also the set of applications A.
- c_k is the *content characterization*—a description of the information content of the transmission. The set of all possible content characterizations is C.
- u_k is the *flow characterization*—a function that defines the owner's quality of service requirements for the transmission. The set of all possible flow characterizations is U.

A policy provides, for each request r_k, an importance value i_k. The importance is a function of the request's owner, source, set of destinations, and content characterization:

$$i_k = p(w_k, s_k, \mathbf{d}_k, c_k)$$

The set of possible importance values is I, and has been defined as the set of real numbers between 0 and 1. More formally, a resolved policy is a function $p : W \times A \times \mathbf{P}(A) \times C) \to I$, where $\mathbf{P}(A)$ is the power-set of A (the set of all subsets of A).

Policy Conformance Metrics—IPSO-FACTO uses the set of policies captured from commanders to assign an importance value to any incoming request for communication resources (illustrated conceptually in Figure 1). Each incoming request is associated with a commander's governing policy and then the request is matched against the sequenced series of policy element statements that commander has made. The first policy element that matches the request determines the importance of that request. At present, contents and legal request syntax are implemented for equality and “descendant-of” relationships, but a future implementation will allow partial and ‘best’ matches.

Importance is shown in Figure 1 as a single numerical value. Of course, not all requests can be satisfied exactly as requested. Rather than rejecting these requests outright, or perhaps equally bad, satisfying them in whatever manner the allocation mechanism chooses, we have posited a set of quality of service dimensions for information. Thus, importance more generally is a shaped function with different profiles indicating how important it is to match a requestor’s *stated* need with *delivered* service along any of a number of service quality dimensions. The dimensions we are currently using include:

- **F**reshness – how current the delivered data is
- **R**eliability – how certain the data is to be delivered
- **I**nitiation-time – the time by which the data must be delivered (first delivered for periodic requests)
- **A**ccuracy – the correspondence of the data to ground truth
- **R**esolution – the grain size of the data
- **S**cope – the “region” over which data is to be provided

The requestor can provide information on how useful the request would be if satisfied suboptimally in any given FRIARS dimension. There is some tradeoff between the complexity of the “utility reduction functions” and the ease with which AIC can manipulate them to optimize request satisfaction; we limited the functions to piecewise linear representations for this year.

The meaning ascribed to importance by the commander, IPSO-FACTO and AIC must match (or at a minimum, be compatible) for the resulting allocation to satisfy the commander’s intent. AIC treats resource allocation as an optimization problem, and attempts to maximize some measure of *total information delivery value*. This value should estimate the contribution of the request satisfaction to the likelihood of mission success. There are various ways to estimate this information delivery value. A simple approach would be:

$$value = \sum_k i_k \cdot u_k(QoS_k) \tag{1}$$

where i_k is the importance that IPSO-FACTO assigns to request r_k, QoS_k is the quality of service assigned to the request, and u_k is a flow characterization—the function that defines the utility of this quality of service to the owner.

Another measure of total information delivery value is delivered value as a function of importance. This measure encourages full satisfaction of requests with a particular importance i before allocating any resources to requests of lower importance. It models the intent of a commander who wants the most important information to get through, even if that prevents many medium-important requests from being fulfilled. Mathematically, value is not a scalar value but a function $value : I \to \mathbf{R}$:

$$value(i) = \sum_{i_k = i} u_k(QoS_k) \tag{2}$$

AIC’s optimizer can compare the values of any two resource allocations using the following order relationship on the value functions:

$$value_1 > value_2 \Leftrightarrow \exists i \in I \ (value_1(i) > value_2(i) \wedge \\ \forall i' > i \ value_1(i') = value_2(i'))$$

In other words, $value_1 > value_2$ if the total utility delivered under the two resource allocations are equal for each importance value down to some level i, and the first resource allocation delivers greater utility at importance level i. This value measure is used by at least one of the AIC implementations.

Multi-user Policies—In realistic military operations, there is never a single commander who gets to make decisions about resource usage. Rather, each commander must allocate his/her resources in accordance with the policies of those above. We support this requirement (Figure 2) by modeling policies that exist at nodes in a command hierarchy. As requests come in, they are matched against the commander’s policy that governs them, but must then also be matched against his/her commander’s policy—and so on, up the chain of command. We allow each commander to stipulate how this matching policy element should be resolved with the subordinate commander’s matching policy element: as a ceiling or floor value, or linear combination of the values (including “ignore” and “use”).

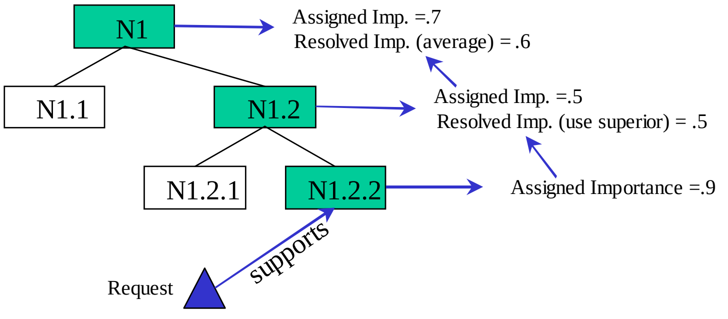
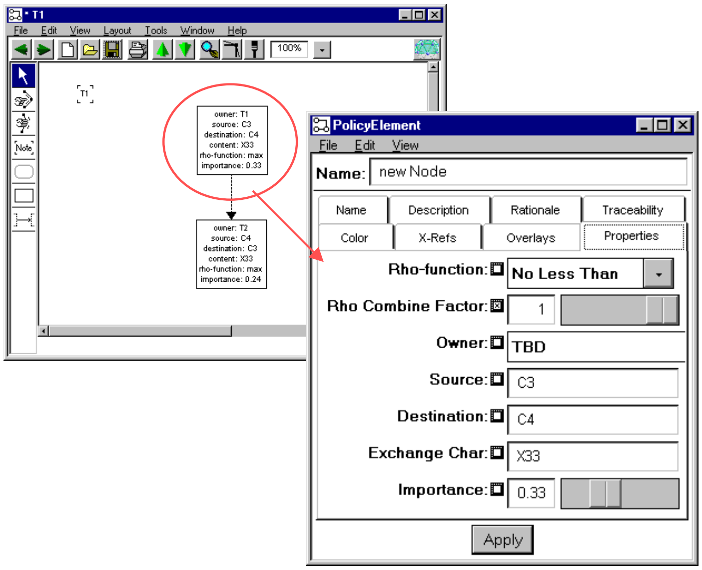


Figure 2. Cross echelon policy application and resolution.

Semi-automated Policy Elaboration and Conversion (SPEC)—We cannot replace the burden of making individual resource allocation decisions with the task of creating numerous policy elements. Information policies are dependent on the commander’s overall goals and mission tasks and the Commander (or his/her staff) currently must interpret the mission plan for its information policy implications. IPSO-FACTO will reduce this overhead by allowing the commander to stipulate the tasks and goals of the mission via ‘task templates’ which reference the information required in the performance of those tasks. Policy implications can be derived from the dependencies and priorities associated with the mission tasks, thereby automatically producing a majority of the detailed intent statements appropriate for the mission.



owner: T1
source: C3
destination: C4
content: X33
rho-function: max
importance: 0.33

owner: T2
source: C4
destination: C3
content: X33
rho-function: max
importance: 0.24

Name: new Node
Rho-function: ☐ No Less Than
Rho Combine Factor: ☒ 1
Owner: ☐ TBD
Source: ☐ C3
Destination: ☐ C4
Exchange Char: ☐ X33
Importance: ☐ 0.33

Figure 3. DOME Policy Creation Tool

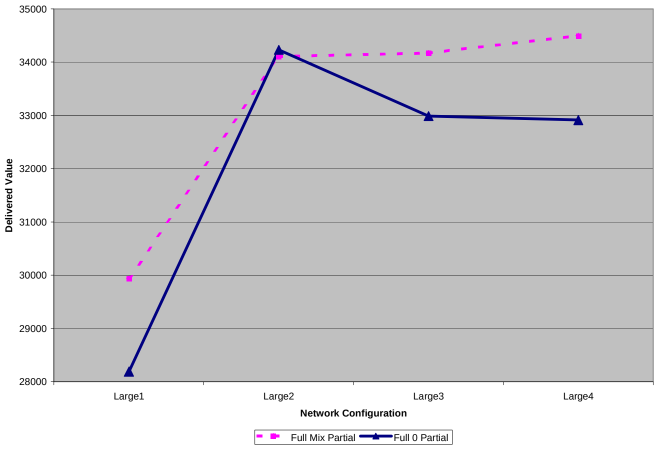
IMPLEMENTATION

In the first year of this effort, we created a prototype suite that allows an off-line user to declare policy elements, specify their order of application, and how subordinate policies should be resolved against them.

The user then transmits the resultant policy to an evaluation engine. The evaluation engine determines the importance value for each request submitted by AIC.

Our year 1 implementation uses a client-server model, with Java used to implement the evaluation server, and a CORBA interface to communicate with the AIC client.

Policies are captured using a graphical software package called the Domain-Modeling Environment (DOME). (DOME is freely available via the web at www.htc.honeywell.com/dome.) DOME then generates an XML file that conveys the policy content to the evaluation server. The policy specification tool provided additional benefit to the experimenter by allowing capture of source, destination, content and scope of authority information.



Our software was evaluated in two scenario settings over a period of four months by TRW in Reston, Virginia, using a test harness they concurrently developed.

RESULTS

The TRW evaluation is still underway for a “militarily realistic” scenario. The other scenario tested was an abstract network configuration for which TRW Reston performed an impressive range of simulation runs, each with extensive logging. With six network variants, thirteen policy approaches, and five load conditions (continuous overload, continuous underload, high, medium and low bursty conditions) there were a total of 390 test runs.

There was a significant difference in the outcomes for the various load conditions. Indeed, in an underloaded scenario there is no benefit to be attained by even a masterful policy-all requests are fulfilled. Thus, we have focused the data analysis on the continuous overload condition, believing it to be the most challenging, and also the most realistic for current network capabilities.

The central question to be answered here is almost certainly: does it make a difference? That is, if we are able to capture the intent of a commander and supply it to some decision-making software, will the results be better than what would have happened absent that intent specification? The core hypothesis we wished to test was simple:

H1: *Allocation guided by policy will outperform allocation without such guidance.*

We expected two other hypotheses to be borne out, each seemingly obvious:

H2: *Policy with added levels of detailed information will outperform more coarse-grained policies.*

H3: *Policy that allows flexibility in the resolution of superior and subordinate importance functions will outperform fixed resolution approaches.*

In fact, in the initial runs we didn’t see any such thing. The reason is simple: the interpretation used by AIC in resource allocation was not the same as the valuation function used by the experimental team to assess outcomes. In fact, using the experimental metric, it can be seen that interrupting a transmission in progress in favor of a (slightly) higher importance transmission is generally a bad idea-the resources allocated to the incomplete transmission have been squandered.

Therefore, we have begun analysis of the experimental runs using a proportional partial credit assignment for incomplete transmissions. This might match a periodic report, or a “resumable” FTP transmission. As seen in Figure 4, assigning partial credit results in the detailed, flexible policy consistently outperforming the policy-free solution. (P(T<=t) < 0.0002, using a one-tailed t-test for paired samples.)

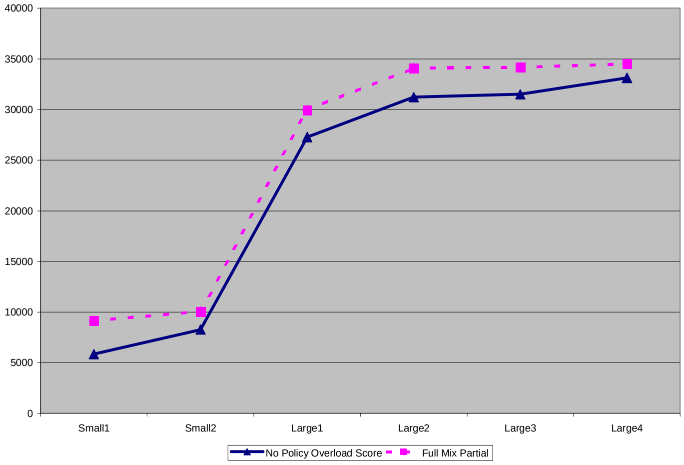


Figure 4. Policy increases delivered value

Given this interpretation of value, the other two hypotheses are supported as well. Figure 5 shows that having more detail (Full > L1, P(T<=t) < 0.04) available to the allocation process improves performance, and Figure 6 shows that using a mixed or flexible resolution strategy() improves performance over a fixed strategy (Full Mix > Full 0, P(T<=t) < 0.02).

While the results of this analysis are quite gratifying, the project itself highlighted a concern we had at the outset-manual specification of policy by a commander would be a tedious and repetitive task. Our current work, which applies this policy-based control guidance approach to aircraft asset allocation also incorporates a task template based approach to semi-automated policy generation. We expect that deriving policy (from more natural and direct statements of what the commander considers important in terms of the mission plan) will substantially reduce the requisite effort.

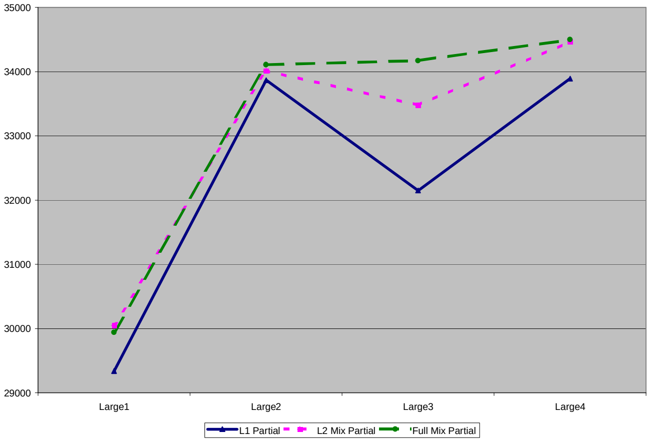


Figure 5. Added detail increases value.

REFERENCES

Funk, H. and Miller, C. (1995). “What do you need to know to know what you need to know: Information requirements to support cockpit information management.” *Proceedings of the 1995 Symposium on Human Interaction with Complex Systems*. Greensboro, NC; September 17-20, 1995.

Miller, C. (1999). Bridging the Information Transfer Gap: Measuring Goodness of Information Fit. *Journal of Visual Languages and Computing, 10*. 523-558.

Shalin, V.L., Miller, C., Geddes, N.D., Hoshstrasser, B.D., Levi, K., & Pershbacher, D. (1990). Automated acquisition of information. *Proceedings of the Soar-90 Conference*. Albuquerque, New Mexico. 1990.

Shattuck, L. (1995). Communication of Intent in Distributed Supervisory Control Systems. Unpublished dissertation. The Ohio State University, Columbus, OH. 1995.

Sheridan, T. (1987). Supervisory Control. In G. Salvendy (Ed.), Handbook of Human Factors. John Wiley & Sons, New York, 1987. 1244-1268.

Vicente, K. J. (1996). Improving dynamic decision making in complex systems through ecological interface design: A research overview. *System Dynamics Review*, 12, 1996, 251-279.

Figure 6. Flexible policy resolution vs. fixed