Chapter 2 – FRAMEWORKS FOR SUPERVISORY CONTROL OF MULTIPLE UNINHABITED MILITARY VEHICLES

Dr. Christopher A. Miller
Smart Information Flow Technologies
211 First St. N. Suite 300
Minneapolis, MN 55401
USA

Email: cmiller@sift.net

2.1 OVERVIEW AND SCOPE

Throughout the meetings of HFM-170, we have discussed the possibility and desirability of having a framework for describing and contrasting the various supervisory control systems that each member group has been working with, as well as others we have experienced. We have discussed many different potential frameworks, without coming to consensus about any specific one of them. The purpose of this document is to review those discussions, along with conclusions reached about the nature of supervisory control frameworks and desirable attributes for this application (as distinct from others), as well as to review the different frameworks proposed and their strengths and weaknesses.

2.2 FRAMEWORKS AND SUPERVISORY CONTROL

Almost concurrently with Sheridan's defining the term and concept "supervisory control", he proposed a framework for characterizing it – his ten stage model which will be reviewed in Section 2.3 below. In Sheridan's definition, which has remained more or less intact since he coined it, supervisory control is any human-machine relationship in which the machine is in a subordinate state *like* that of a human supervisor-subordinate relationship. "… [S]upervisory control derives from the close analogy between a supervisor's interaction with subordinate people in a human organization and a person's interaction with intelligent automated sub-systems." [1].

But just as there is a huge range of human-human supervisory relationships (from master-slave to parent-child to collaborative work team to president-nation to gang leaders), so there are a wide range of human-machine supervisory control relationships and it would be desirable to be able to characterize and discuss their similarities and differences. If for no other reason, it would be helpful to be able to discriminate one from another, especially when attempting to determine which type works best for which application. So, beginning well before Sheridan, there has been a long history of characterizing the types or stages of a human-machine supervisory control relationship.

But it is worth remembering, after George Box [2], that 'all frameworks are wrong, but some frameworks are useful'. Box uttered that quote, he was talking about models, but a framework is essentially a model – and perhaps a taxonomy. Box meant that a model will never fully capture the details and intricacies of the real world, but that some models may capture interesting or relevant aspects of it – and, in fact, by eliminating excess detail, some models may even make it easier to see relevant relationships and distinctions. Similarly, there are always multiple ways of parsing any complex phenomenon into multiple frameworks or taxonomies. None of these will capture the full richness of the phenomenon, but some of them – indeed, multiple versions

– may be useful because they organize the phenomenon in helpful, insightful ways, highlighting some similarities and differences while obscuring others. There is no single "right" framework for describing supervisory control, but different ones may be more or less useful for different purposes. Therefore, for any framework development or evaluation, it helps to begin with a discussion of the purposes or goals which the framework will serve.

2.2.1 Why Have a Framework?

There are a wide set of possible reasons one might want to have a framework for supervisory control. These include:

- *Training* – A framework can serve as a "mall map" for training users in the specific attributes of a single system, or in training the similarities and differences across multiple systems.
- *Interaction* – The "mall map" attributes of a framework can also serve as a mental model to aid users in understanding and remembering the attributes and behaviors a supervisory control system affords.
- *Organizing Investigation* – For research purposes, a framework can serve as a map to "uncharted territory", helping to determine what areas have been investigated and what have not – and can aid in generalizing results if it highlights the similarities and differences between "regions" of the space of possible system alternatives.
- *Understanding Alternatives for Design* – As investigation of the space of alternative approaches to supervisory control are completed, they form a set of data about design alternatives – a database that may be organized according to a framework. Thus, the framework may serve as a guide to designers to understand both what alternative design methods are available and what conditions they have been proven to work well in.

The reasons for this working group were related to the later two – to characterize the set of systems and applications we were discussing as a part of our collaboration. To the degree that the framework we created helped us gain insights from the set of supervisory control applications we studied, that would be added benefit. Finally, we also wanted an organizing principle for this report.

2.2.2 Framework Attributes and Goals

Frameworks can be more or less elaborate, and they can highlight or suppress different aspects of the phenomenon they model. Given the goals described above, desirable attributes for our framework included:

- *Brevity and Simplicity* – To serve as an organizing principle for communicating the results of this working group to the outside world, it was important that the framework be simple and brief enough to be conveyed and explained to a reading audience in a limited amount of time. We were willing to make some sacrifices in the coverage or resolution of the model in order for it to be readily comprehensible by our audience. To some degree, the "fame" (or prior knowledge) of a candidate framework could be used to compensate for simplicity, and some frameworks (most notably Sheridan's [3], [5] and [6]) were already well known in the human factors and engineering world.
- *Emphasis on Domain* – Since this exercise was a part of a NATO RTO working group on supervisory control of unmanned military vehicles, this colored our efforts in three ways:
    - First, "supervisory control" was the focus and topic of our framework development efforts. While many other topics in the domain of human-automation interaction are related to supervisory

control (such as levels of autonomy, adaptive autonomy and user interface and interaction design), these were not directly our focus.

- Second, our emphasis was on developing a framework for characterizing supervisory control *of Unmanned Military Vehicles (UMVs)*. While, as will be seen below, we occasionally discussed other forms of supervisory control (both human-human and other application domains), our ultimate emphasis was on UMVs.
- Third, since supervisory control is, at root, a relationship between human and automation, we focused on characterizing and describing alternate ways in which that relationship could exist. Technology and its application domains, while relevant, were not the primary focus.

- *Coverage* – It was important that the framework developed be able to cover – that is, to represent and describe – each of the applications that were being developed and tested by the various participating members and their countries' laboratories. Coverage outside that set was nice to have, but deemed less important.
- *Distinction* – While the ability to cover the set of demonstrations being developed by the working group participants was important, it was equally important that our framework be able to make distinctions between them. We wanted to be able to organize and characterize these demonstration systems, identifying their similarities and differences.

2.3 ALTERNATE FRAMEWORKS CONSIDERED

In this section, we will describe the various frameworks considered during the course of the working group.

Sheridan

Sheridan's initial "framework" for characterizing supervisory control relations was a spectrum arrayed as a 10 item list whose endpoints are full control autonomy for the human (essentially no role for automation) and vice versa [3]. The intermediate levels in this spectrum, then, represent alternative forms of supervisory control interactions. A version of this list is shown in Table 2-1.

Table 2-1: Levels of Automation (after [3]).

1	Human does it all.
2	Computer offers alternatives
3	Computer narrows alternatives down to a few
4	Computer suggests a recommended alternative
5	Computer executes alternative if human approves
6	Computer executes alternative; human can veto
7	Computer executes alternative and informs human
8	Computer executes selected alternative and informs human only if asked
9	Computer executes selected alternative and informs human only if it decides to
10	Computer acts entirely autonomously.

Strengths and Weaknesses

Sheridan's spectrum is very simple and very well known, but it has several flaws as a framework for our needs. It has been criticized (in Parasuraman, Sheridan and Wickens [6], among others) for confounding automation employed in the presentation of information, in the making of decisions and in the executing of actions. Because most military systems are complex enough to employ automation operating on many different tasks and task types concurrently, when one assigns a number using Sheridan's framework, one is forced to either be ambiguous about the task or operational domain being described by the number, to break tasks down to a fine enough level where only one of Sheridan's levels makes sense as a descriptor, or to "average" over the automation levels and the tasks and applications where automation is provided. This argument is made in more detail in Miller and Parasuraman [4],[9].

Parasuraman, Sheridan and Wickens

Sheridan's spectrum model is essentially uni-dimensional. As noted above, though, it achieves "unidimensionality" by combining several potential behaviors that automation can perform. While extremely simple and intuitive, this unidimensionality may well be too simple to make the kinds of distinctions between systems and applications which we would like to make.

Another problem with uni-dimensional models of human-automation relationships is that they are ambiguous about what the application domain of the relationship applies to. Parasuraman et al. [6] noted that Sheridan's levels referred mainly to automation that makes decisions, offers suggestions and/or executes actions. There are, however, other jobs automation can do: for example, sensing and analyzing information to detect situations of interest, without necessarily offering any advice on what to do with the information. Parasuraman et al. [6] applied a simple, stage model of human information processing to arrive at four functions that must be accomplished to perform most tasks:

1) Information acquisition;
2) Information analysis;
3) Decision and action selection; and
4) Action implementation.

Since these functions can be performed by either human or automation in various mixes, in effect Parasuraman et al., [6] added a second dimension to Sheridan's spectrum – that of the function or task the relationship is defined over. Most human + automation systems can be characterized by a mix of LOAs across these four functions, as in Figure 2-1. One system (A) might be highly autonomous in information acquisition, but comparatively low on the other functions, while a second (B) might offer a high LOA across all four functions.

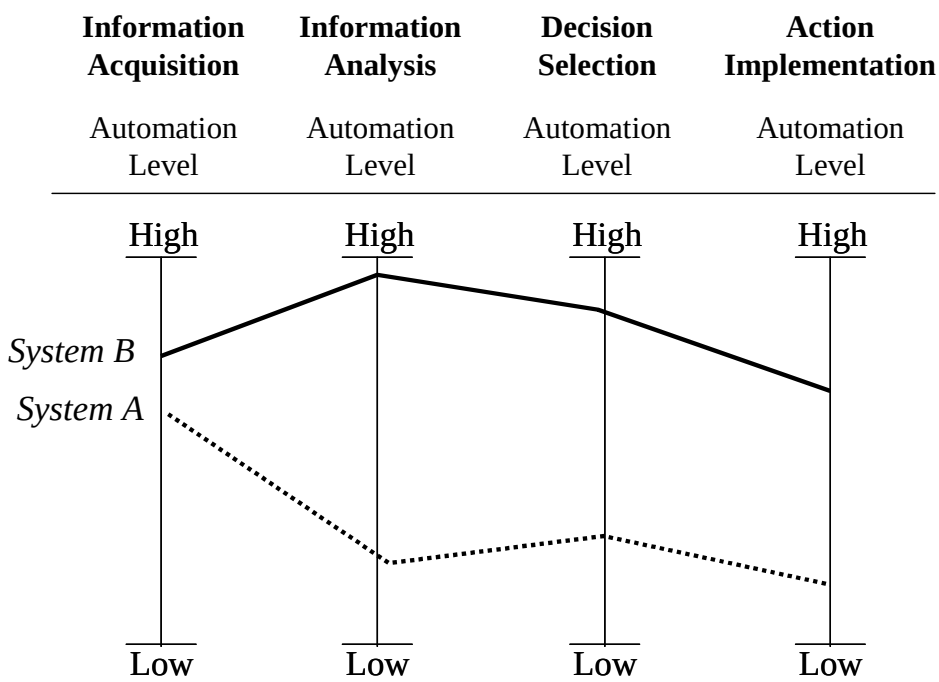


Figure 2-1: Levels of Automation by Information Processing Phase for Two Systems (from [5]).

Strengths and Weaknesses

This information processing stage ✕ autonomy level model has gained extensive acceptance in the research community and systems have begun to be developed in accordance with it. Research has shown different effects of automation at the different stages [7]. Nevertheless, it was felt that it too was too coarse grained to provide adequate distinctions between the various supervisory control systems being explored by members of HFM-170. Furthermore, some members of the group had reported prior attempts to use the Parasuraman, Sheridan and Wickens framework to describe and define systems and had encountered difficulties in making clean distinctions between the information processing stages.

2.3.1 Miller and Parasuraman

An implication of the Parasuraman et al. [6] levels x functions model is that a parent task[1] can be decomposed and that a single automation level need not be applied homogenously across the sub-tasks. However, in their model a parent task is decomposed into abstract sub-functions based on information processing stages, whereas other decomposition methods might arguably provide more insight into how a task may be performed. In fact, the role of task analysis in Human Factors is to perform exactly such decompositions in a

[1] Note that in Sheridan's model, as well as in Parasurman, Sheridan and Wickens, and all of the other models discussed in this document, a decomposition of the functions to be performed by human(s) and automation is important, but it is less important whether the focus of analysis is prescriptive tasks, abstract functions or even the goals which are accomplished by those tasks and functions. Since supervisory control necessarily concerns itself with allocating the work of multiple agents for the accomplishment of a goal, allocation will necessarily include goals, functions and the tasks or methods which accomplish them. Most of the models described in this document apply regardless of whether the supervisor allocates via prescriptive tasks or ecological abstract functions and goals.

hierarchical fashion through any number of levels to some primitive, "stopping" level [8] that may be imposed by biology, physics or, more commonly, the purpose of the decomposition.

In a 2003 paper, Miller and Parasurman [9] argued that while the two-dimensional LOA model offered by Parasuraman et al. [6] represents a major advance over earlier uni-dimensional models, it arguably does not go far enough. The subdivision of a parent task into four information processing phases represents only a single level of decomposition into abstract task categories. In practice, tasks are accomplished by hierarchically decomposable sequences of specific activities – the parent task's sub-tasks. Automation may be applied differently to each and every sub-task that comprises the parent task. Thus, the profile of automation levels sketched in Figure 2-2 could stretch instead over as many sub-tasks and levels as we want or need to divide a parent task into.

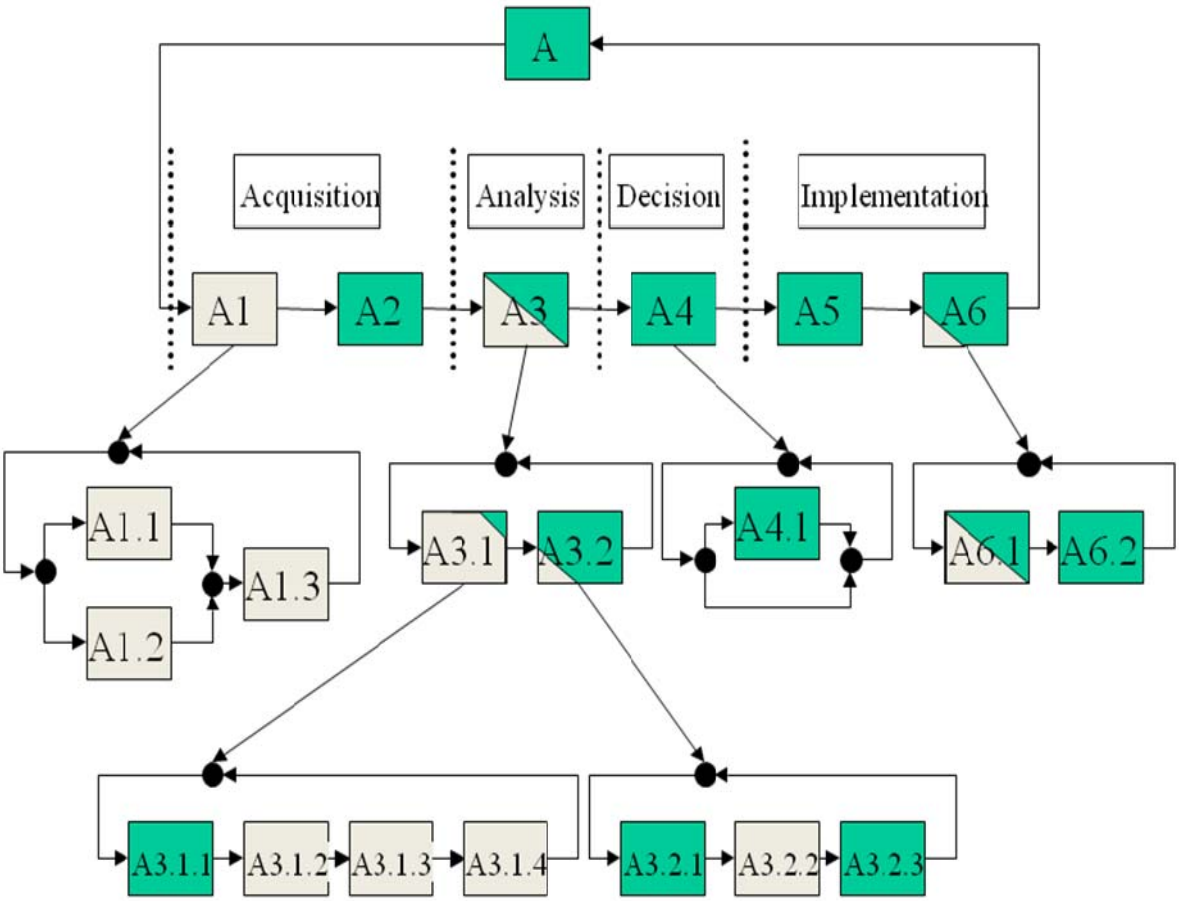


Figure 2-2: Hypothetical Decomposition of Task A Into a Hierarchical Set of Sub-Tasks – Each of Which May Have Differing Automation Levels.

In fact, the relationship between automation "level" and task decomposition is more complex still. As is well understood [6], automation does not merely shift responsibility for tasks but can change their nature as well. In a task decomposition, this means that some sub-tasks may be eliminated while others are added. This implies that there will generally be multiple alternate decompositions depending on, among other things,

what LOA is used. Each alternative constitutes a different combination of human and automation sub-tasks and, thus, a different method of accomplishing the parent task.

When one identifies a LOA for a complex system using Sheridan's dimensions, one is in principle identifying something like an average or modal level over the sub-tasks the human + automation system accomplishes. Similarly, when one uses a levels × stages model such as in Parasuraman et al. [6], one is performing an abstract and coarse-grained decomposition of the parent task into sub-tasks clustered by information processing stage. Assigning levels by sub-task stages offers more sensitivity than assigning them only to the parent task, but it is still an abstraction. In practice, one could identify the specific sub-tasks to be performed and represent an automation level for each of them.

Figure 2-2 provides a hypothetical illustration of this relationship. A parent task "A" might be said to have a certain level of automation on Sheridan's scale, but in fact, that task is comprised of a series of sub-tasks (A1 – A6) each of which may have a different mix of human and automation involvement (as illustrated by the different shadings of the sub-task boxes). These sub-tasks can be reasonably organized or clustered into Parasuraman, Sheridan and Wickens's information processing stages, but this can obscure both the fact that specific sub-tasks exist and that they may have different mixtures of human and automation involvement. Furthermore, each of the tasks at this level can also, generally, be further decomposed into sub-sub-tasks (A1.1 – A6.2) which may also have different mixes of human and automation involvement… and so on, until some desired primitive level is reached.

Why would one want to perform this kind and level of analysis? More and finer-grained sub-tasks are not necessarily better and, in fact, Parasuraman et al. [6] explicitly state that they chose a four-stage model to simplify design considerations. Precision may be inherently desirable for some purposes (such as training and detailed design), but Miller and Parasuraman's [9],[4] purpose in achieving greater precision and finer granularity was to support flexible task delegation. As we saw above, for any intermediate LOA for a task, there are roles for both humans and automation in its sub-tasks. Yet, someone must coordinate those roles. Insofar as human supervisors are required to manage, or at least be aware of, that division of labor, they must understand the decomposition of the task and of the allocation and coordination requirements among its sub-tasks. Supervisory control is a process of task delegation and delegation requires task decomposition.

Strengths and Weaknesses

That said, while precision and flexibility in stipulating how a task is to be performed is a necessary aspect of powerful delegation relationships, the goals of a framework for this working group were somewhat different. Instead of precisely defining each and every difference in how various systems achieve supervisory control, we wanted to group similar systems – a process that implies some degree of ignoring (or, at least, clustering) differences. Furthermore, prior work with the Miller and Parasuraman [9] approach (as well as the similar LoA^3 and CLAMP^3 frameworks described below) has shown that, while powerful, they are as cumbersome and time consuming to use *for the purposes of description* as most task analytic techniques in human factors [8],[10]. Thus, they were inappropriate for the purposes of briefly and intuitively summarizing the similarities and differences between supervisory control systems being explored in the HFM-170 technology development efforts.

2.4 LoA^3

During prior work for the U.S. Air Force Research Laboratory and in a proposal to the U.S. Army's Aeroflightdynamics Directorate (AFDD), Miller et al., expanded the concept of hierarchical task decomposition

to describe supervisory control to develop what they called the "LoA[3]" concept. This stood for levels of authority, abstraction and aggregation, and this triumvirate of parameters was advanced as a way of describing and defining delegation relationships. Delegation, in general, was defined as illustrated in Figure 2-3.

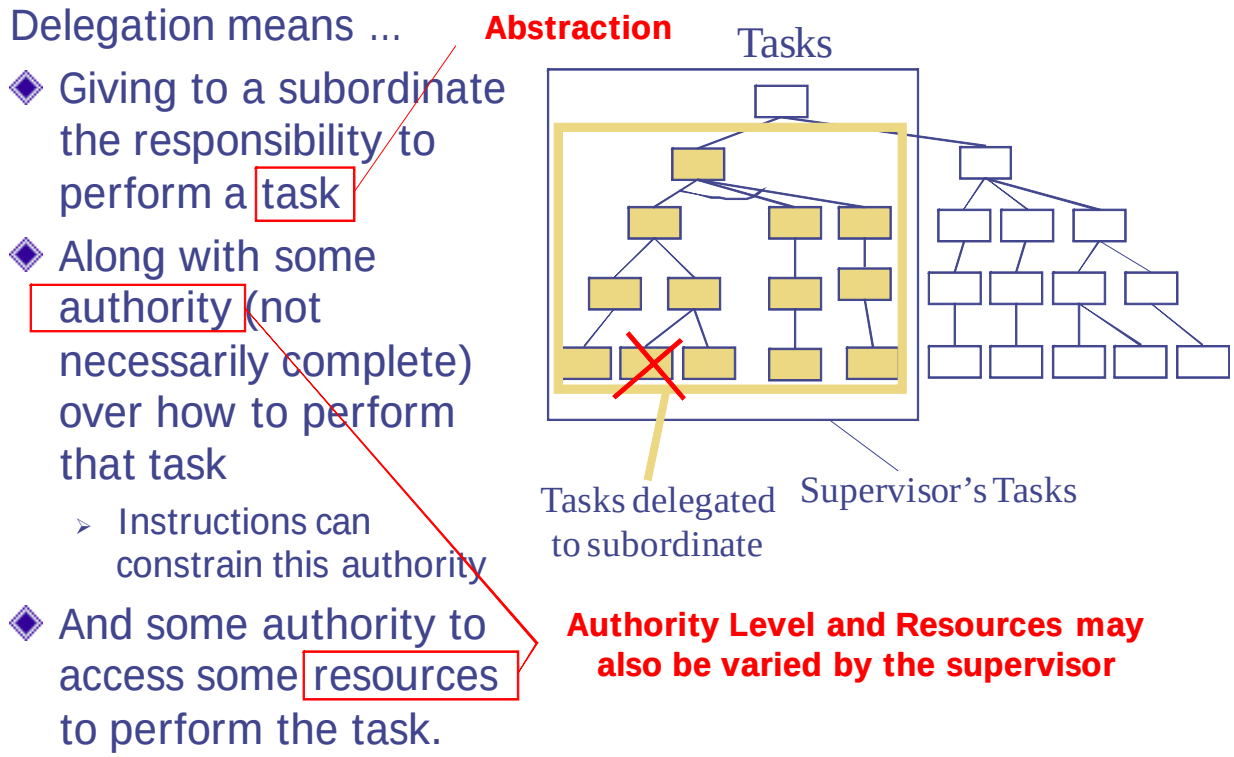


Figure 2-3: A 3-Dimensional Definition and Framework for Delegation Interactions.

The three dimensions were:

1) *Abstraction* – This is essentially the means-ends or hierarchical decomposition of a task. Each parent task represents the "ends" or goal of a child (or set of child) tasks; each child task represents the means by which a parent task is accomplished. Delegation means handing over some responsibility for some tasks which themselves are part of larger, parent tasks and which also decompose into smaller child- or sub-tasks. Delegating the task, at any level, means transferring *some* authority to perform that task (if any workload is to be saved) and authority over some resources (at least attentional resources) to perform the task and its sub-tasks and to make the decisions about which sub-task methods to pursue.

2) *Aggregation* – This is essentially the part-whole decomposition of resources. Any effective act of delegation must include the delegation of (at least partial) control over resources if it is to accomplish anything – even if the resources delegated are only the attentional and decision making resources of the subordinate. Of the full set of resources which a supervisor controls, s/he will delegate some control over some of those resources to the subordinate.

3) *Authority* – Authority represents the degree of autonomy that the subordinate has over the tasks (abstraction dimensions) and resources (aggregation dimension) he/she/it has been delegated.

Sheridan's 10 levels (cf. Figure 2-1 above) can be thought of as a spectrum of autonomy – characterizing a range from "computer has no autonomy" through varying levels of autonomy to provide suggestions only, autonomy to act only if authorized to do so immediately prior, authorized to act only if the action is reported, and ending at "computer has full autonomy".

These three dimensions, then, let us specify an act of delegation as depicted in Figure 2-4. Any delegation involves the transfer of responsibility for some sub-tasks or tasks for which the supervisor has responsibility, along with some resources over which the supervisor has control. In both cases, the transfer of authority may not be complete, even over these sub-tasks. The supervisor may require the subordinate to coordinate, ask permission, inform, etc.

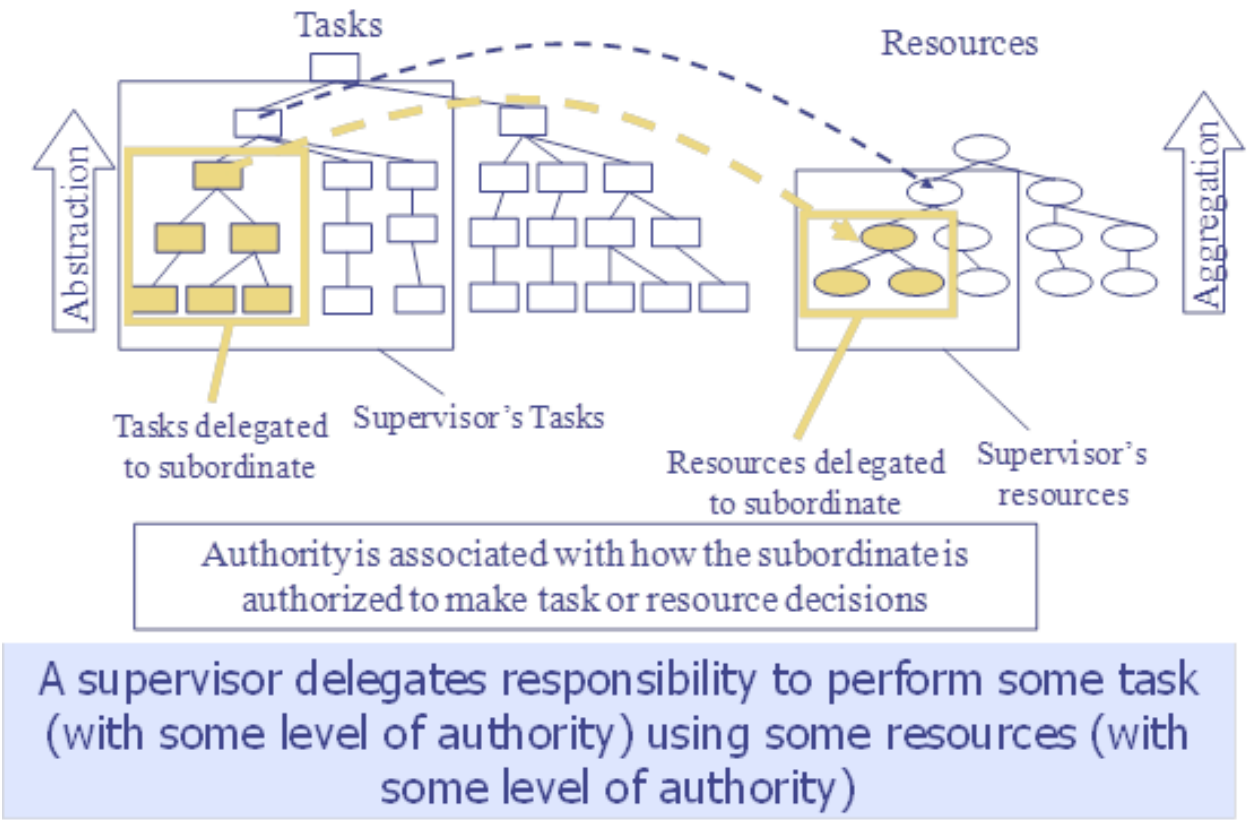
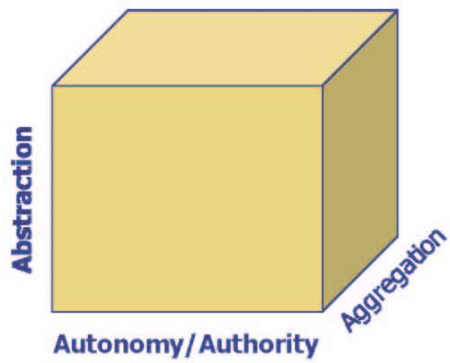


Figure 2-4: An Act of Delegation Can Be Specified as a Transfer of Responsibility Along Three Dimensions.

Note that if the delegation act extends over time and functions, it establishes a *supervisory control relationship* between a supervisory and a subordinate. Therefore, the three dimensions of the LoA^3 model define a three dimensional "space" within which we could place, in principle, any supervisory control relationship (or act) – as illustrated in Figure 2-5.



- Level of Abstraction– At what level(s) can the human command; in terms of task hierarchy– with what level of authority?
- Level of Aggregation– what resources (how many agents/systems) can the human command– with what level of authority?
- Level of Autonomy or Authority– How independently can the automation act? Sheridan's spectrum

These are dimensions that can be varied in LoA experiments

Figure 2-5: A 3-Dimensional "Delegation Space" Formed by the LoA^3 Dimensions.

Strengths and Weaknesses

The LoA^3 model provides a rich description of the *act* of delegation, but there were problems with it for the purposes of our technical group. First and foremost, it relies on the same hierarchical task decomposition as the Miller and Parasuraman approach described in Section 2.3.1 above. As discussed there, this approach essentially requires a detailed and complete task decomposition for each alternative considered. While that may be useful for design and for a deep understanding of system alternatives, it is not compatible with the easily understood taxonomic groupings this working group was after. Second, this model was seen as having a failing in that it concentrated *exclusively* on the delegatory act and had nothing to say about the environment or technological context in which that delegation was performed. Since the members of HFM-170 were working on characteristically different technologies and domains (e.g., ground vs. air vs. sea UMVs), it was felt that an adequate framework for the group needed to reflect these differences and similarities as well.

2.5 CLAMP^3

CLAMP^3 was an attempt to remedy the second failing of the LoA^3 model as described above – to embed the LoA^3 model in a broader framework which would include the ability to characterize aspects of the technology and application domains of the supervisory control systems it classified. CLAMP^3 was developed by Harry Funk and used as the framework for a simulation and testing environment for delegation systems which was built and used initially by Jay Shively, Susan Flaherty and Lisa Fern at the U.S. Army's Aeroflightdynamics Directorate (AFDD) and later expanded and used for additional experiments by personnel at Smart

Information Flow Technologies (a small U.S. business) and George Mason University under funding from AFDD.

CLAMP^3 stands for C^3 (three dimensions of Context) + LoA^3 + Mapping for Predicting of Personnel Performance – as illustrated in Figure 2-6. In other words, CLAMP^3 takes the 3 levels describing the delegation interaction from LoA^3 and "wraps" them in a description of the context in which that delegation action takes place (the three context dimensions) along with a description of the resulting performance metrics for the human-machine system.

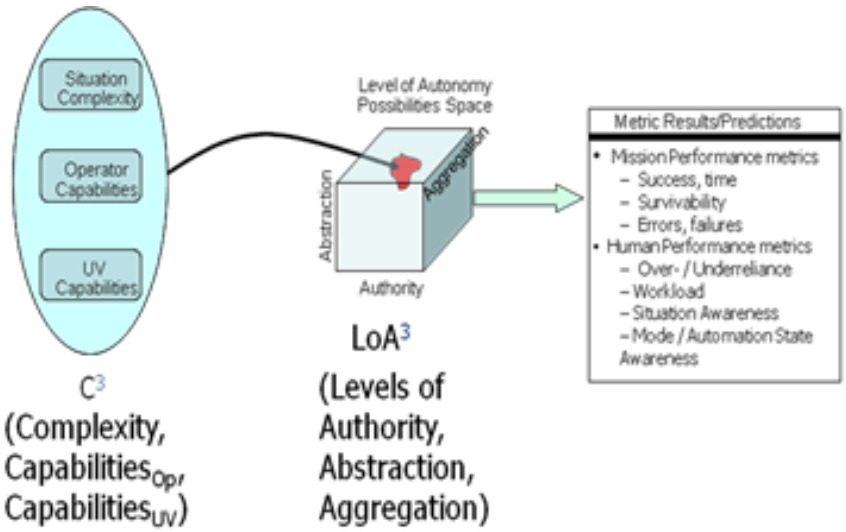


Figure 2-6: The CLAMP^3 Model – Embedding LoA^3 in Context and Performance Outcomes.

The three context dimensions used were:

1) A description of the *Situation Complexity* – That is, for example, is the UAV being asked to fly straight and level at cruising altitude on a clear and windless day, or is it being asked to fly nap of the earth at night, in storms, with enemy radars and small arms fire.

2) A description of the *Capabilities of the Operator* – Is s/he a trained fighter pilot (training that might be helpful for operating a UAV, but useless or even counter-productive for operating a UGV) or an untrained infantryman? Is s/he operating in a quiet room devoting full attention to the UMV management task, or is s/he engaged in combat, taking fire and perhaps riding in the back of an armored vehicle in rough terrain?

3) A description of the *Capabilities of the Unmanned Vehicle* – These could be any relevant attribute of the UMV, but particular relevance will generally be associated with control and functional capabilities. A vehicle which only ever does one thing (e.g., flies in a circle transmitting images) will impose much less burden on an operator than one that admits many different behaviors and modes. Similarly, one whose performance and stability is unreliable will require much more human attention than one that is highly reliable.

At the "other end" of CLAMP^3 is a description of the outcome or effects of a delegation relationship within the context described – that is, performance measures in terms of both mission and human performance.

Strengths and Weaknesses

While CLAMP^3 has been used as the framework for experimental work sponsored by AFDD, it retains problems for use as a framework for this working group. First, it inherits the problem of LoA^3 and of Miller and Parasuraman [9] in that it is based on a hierarchical task decomposition to describe the delegation relationship between human and automation. While this is rich and detailed, it is likely too rich and detailed for convenient use by this group or easy understanding by others.

More importantly, while CLAMP^3 calls attention to the need to situate a description of a supervisory control relationship in a context and to describe its effectiveness, methods of representing these dimensions are underspecified. CLAMP^3 tells us, for example, that it's important to consider the complexity of the context, but gives us no metric or even set of factors that might contribute to that complexity. Such would be necessary for comparing applications within and across the members of HFM-170. The main contribution of CLAMP^3 was to remind us to include these dimensions as we moved forward in trying to specify a framework for use by this group.

2.6 7D

There was general consensus that the core ideas of the CLAMP^3 framework were moving in the right direction – that is, particularly the idea that folding a description of a supervisory control relationship into a description of the context in which the relationship was used to describe the resulting system. What was needed, it was felt, was a way of reducing the complexity of the associated dimensions while regularizing and scaling them. It was with these goals in mind that Chris Miller proposed a seven dimensional model at the Stockholm meeting of HFM-170 in June, 2008.

The seven dimensions of this model were formed by returning to the LoA^3 and C^3 (context) dimensions of the CLAMP^3 framework and attempting to characterize and scale them to develop a multi-dimensional description of a supervisory control relationship plus the environment in which that relationship occurred. More specifically, the goal was to use the previously-identified abstract dimensions to form a quantified and specific set of relevant and important dimensions along which the HFM-170 projects and applications varied.

The LoA^3 scales (autonomy, abstraction and aggregation) collectively characterize the *interaction* between the human and automation, as we noted above. The C^3 scales (complexity, operator capabilities, automation/UMV capabilities) collectively characterize the *environment* in which the LoA relationship exists and is exercised. Using this insight, Dr. Miller went through each of the scales and attempted to distill what was "important or significant" about each of them with regard to the supervisory control applications being developed and discussed by this working group. This gave us a means for identifying dimensions to include in our resulting model and, more importantly, for creating a scale for each dimension. The set of dimensions identified are depicted in Figure 2-7 (which also illustrates how they were derived from the LoA^3 and C^3 dimensions) and are discussed along with the scales developed to represent them below.

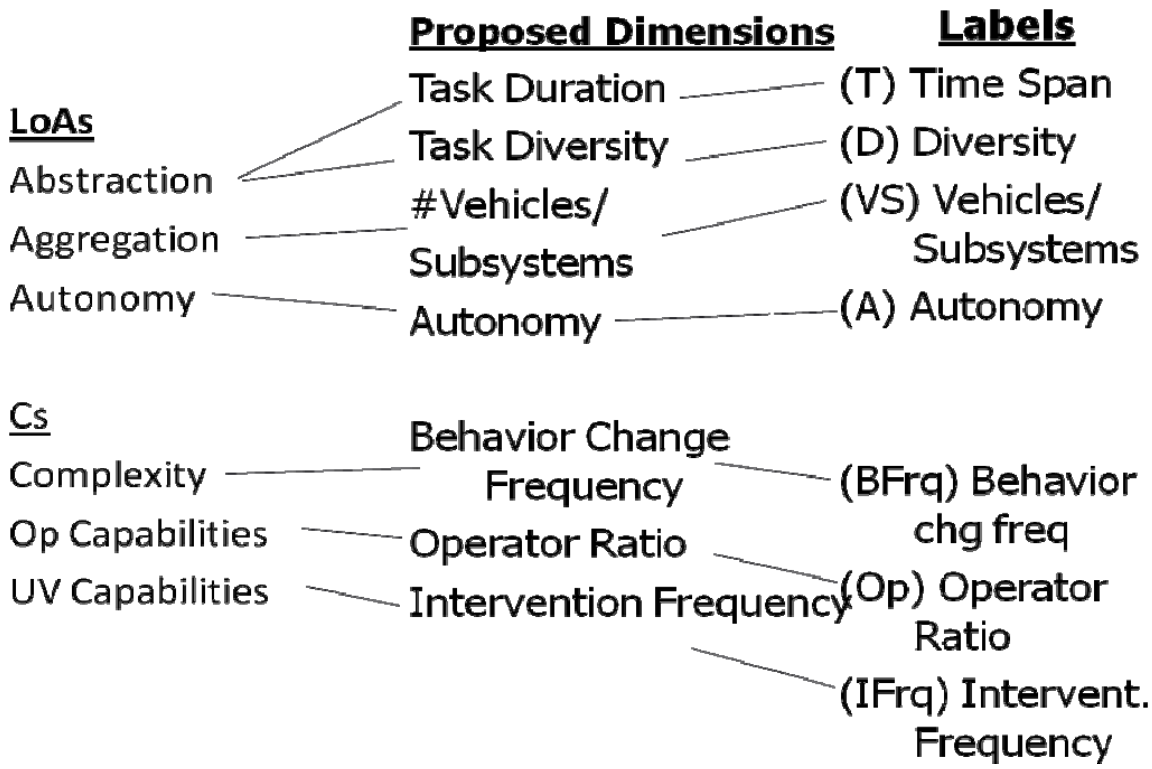


Figure 2-7: The Seven Descriptive Dimensions and Their Derivation from the LoA^3 and C^3 Dimensions of the CLAMP^3 Model.

In general, the scales were created to capture the range of variation, yet show interesting degrees of difference, along the dimensions we saw in supervisory control UMV systems under consideration by the group. A secondary motivation was an attempt to synchronize the length and scalar values on each of the scales to facilitate later visualizations. To achieve this later goal, we developed seven point scales for each dimension (as described below) and arrayed each of the scales from "worse" (notionally less competent systems) to "better" (notionally more competent systems). The specific divisions are, of course, somewhat arbitrary and debatable, but the intention was to provide a "chunking" of the dimension into seven significantly different categories:

- What's important/significant about "Abstraction"? Abstraction, in a task hierarchy sense, captures the number, types and relations of tasks/behaviors the UMV is designed for. If the top level task in a hierarchy for a given UMV can be thought of as "Perform Mission", then a complete decomposition will represent all the possible tasks that system will perform. The "size" of that hierarchy tells us important things about the mission(s) and capabilities of the UMV and led to proposing two different dimensions:
    - **Mission/Task Duration (T) –** Duration of missions is a reasonable stand-in for "size" of the hierarchy – how much time span does a typical mission or task being analyzed cover? This is not the task that the operator delegates, but rather the operational window for the UMV itself. Length/duration is a simple dimension ranging from seconds to days or weeks. A scale of interestingly different levels on this dimension (for the set of UMV systems we were considering) was proposed as:

1) Seconds
2) 1 – 5 minutes
3) 5 – 30 minutes
4) 30 – 90 minutes
5) 1.5 – 6 hours
6) 6 – 24 hours
7) Days

- **Task Diversity (D) –** Task diversity is a necessary second dimension to identify the "complexity" of missions and of UMV roles. How many different *types* of tasks is the UMV involved in? How many conceptually distinct functions[2] are performable by the vehicle(s)?:
  1) 1 only
  2) 2 – 3
  3) 4 – 6
  4) 7 – 10
  5) 11 – 15
  6) 16 – 25
  7) 25+

- What's important about "Aggregation"? Aggregation refers to the number of vehicles (or vehicle "parts" or sub-functions) being controlled by the user(s) in an application to be characterized by the framework. Some supervisory control systems are being designed to control multiple UMVs simultaneously, while others are controlling at most a single sub-system. This gave rise to the **Vehicles/ Sub-systems** (VS) dimension.
  - **Vehicles/Sub-Systems (VS) –** The VS dimension captures how many UMVs and/or UMV sub-systems are typically involved in a mission (in the analyst's focus of interest):
    1) Single sub-system
    2) Multiple (2 – 4) sub-systems, but not whole vehicle
    3) One whole vehicle or 4+ sub-systems
    4) 2 whole vehicles (or parts thereof)
    5) 3 – 4 vehicles
    6) 4 – 12 vehicles
    7) Swarms (12+)

- What's important about "Autonomy"? Who's in charge, who is leading/following? For the mission as a whole, what's the relationship between human and automation?
  - **Autonomy (A) –** To characterize this dimension, we relied on an abbreviation of Sheridan's initial autonomy scale which folds in a sense of where, in the hierarchy of tasks which comprise the mission, the control is taking place:

[2] Of course, determining what a "conceptually distinct function" or task type is will be subject to individual judgment and to the needs and focus of the analysis. This is largely irrelevant as long as the selected level is kept approximately constant across systems to be compared.

1) Human is in charge and commands specific, limited, non-integrated functions from automation
2) H sets overall goals, dictates tasks, but delegates moderate decision authority within isolated functions to A, while retaining monitoring and intervention authority
3) H responsible for overall goals, but A is given large tasks which may integrate across functions. A may initiate actions within its functions
4) Balanced responsibilities between H and A
5) As for 3, but switch H and A
6) As for 2, but switch H and A
7) As for 1, but switch H and A

- What's important about environmental "Complexity"? We argued that this could be captured by noting how often the UMV has to change its behaviors (where "behaviors" are significant variations within the tasks or functions defined for "Task Diversity" above). This dimension was called "Behavioral Change Frequency" or BFrq.
  - **Behavioral Change Frequency (BFrq) –** What is the average duration between required changes in vehicle behaviors (either user- or system-initiated) in a typical mission of interest?:
    1) Longer than 1 per hour
    2) Every 20 – 60 min
    3) Every 5 – 20 min
    4) Every 1 – 5 min
    5) Every 10 – 60 sec
    6) Every 5 – 10 sec
    7) Once per second or faster
- What's important about "Operator Capabilities"? Here, we felt that all the required operator capabilities, while significant in their own right for training and selection, etc., could be rolled up and reflected in how many operators are required to control the vehicle(s) in a typical mission on which the analyst is focusing – hence, Operator to Vehicle Ratio (Op).
  - **Operator Vehicle Ratio (Op) –** In the scale developed below, ratio can be calculated – thus, four operators controlling four UMVs yields a ratio of 1:1. Similarly, fractions of an operator's time may be considered if the operator is concurrently engaged in other tasks – thus, an infantry soldier who is spending half his time controlling and monitoring video feed from a UAV while providing covering fire could be represented as .5 operators to 1 UMV or 1:2.
    1) 4+ operators to 1 UV
    2) 2 – 3 Ops to 1 UV
    3) 1 to 1
    4) 1 Op to 2 UVs
    5) 1 Op to 3 – 4 UVs
    6) 1 Op to 5 – 10 UVs
    7) 1 Op to 10+ UVs
- What's important about "UMV Capabilities"? Here, we argued the raw capabilities of the UMV were not as important (and were too diversified for good abstraction in a model), but rather its capabilities

to perform its functions without operator intervention. This, in turn, gave rise to a focus on the frequency with which the operator had to intervene in the functioning of the UMV(s).

- **Operator Intervention Frequency (IFrq) –** This dimension was captured in a scale tied to the required frequency with which the operator had to interact with the system to achieve successful mission behavior. (Note that later thinking, not adopted at the time of presenting and discussing this dimension, suggests that this should not be a simple intervention frequency, but rather a percentage or ratio of clock time which the operator must spend in interaction with the vehicle – similar to "Robot Attention Demand" in Olsen and Goodrich's (2003) "fan out" metric.):
    1) Once per second or faster
    2) Every 5 – 10 sec
    3) Every 10 – 60 sec
    4) Every 1 – 5 min
    5) Every 5 – 20 min
    6) Every 20 – 60 min
    7) Longer than 1 per hour

Once these scales had been developed, we sought to illustrate them on a set of examples drawn from significantly different UMV systems. We chose a set of three systems:

1) *Unattended Ground Sensors (UGSs)* – These represented a very simple (perhaps degenerate) example of UMV systems. A UGS is a simple sensor which, once installed (or even air dropped), transmits a video or auditory signal only when a stimulus of interest (e.g., a heavy vehicle passing by) is detected. Tens or perhaps hundreds of UGSs can be installed in an area and "controlled" by a single operator. Once installed, they do not move or change their behaviors except transmitting vs. not transmitting.

2) *Raja's RoboFlag* – In 2003 and 2004, Dr. Raja Parasurman and others worked with a simulated robotic platform created by Dr. Mark Campbell of Cornell University to conduct a series of experiments in flexible, delegation-style supervisory control [11],[12],[13],[14]. In this testbed, a single human operator controlled a team of five ground-based robots maneuvering them about a playing field to play a game of "capture the flag" against a fully automated team of five opposing robots. The robots could move, sense other robots, "tag" other robots to disable them and grab the enemy's flag and return it to their territory. The operator could control these robots (in varying experimental conditions) by either individual waypoint commands, or a series of increasingly aggregated "plays" which might task a single robot or the entire team.

3) *Jay Shively's MUSIM and Delegation Control (DelCon) Environment (a.k.a. "Jaybook")* – DelCon [15],[16] is a flexible delegation-style supervisory control system being developed by Jay Shively and colleagues at the U.S. Army's Aeroflightdynamics Directorate. Delcon (or, as it was affectionately known by members of the working group, "Jaybook") is embedded in the Multiple UAV Simulation (MUSIM) testbed, where it controls three UAVs in the performance of an urban target monitoring, search, tracking and prosecution scenario. Jaybook provides control capabilities that range from waypoint control and joystick-controlled sensor operations to multiple UAV coordinated monitoring and lasing/prosecution plays.

Using the scales defined above, these three example systems can be graphed to illustrate the characteristic differences between them. Figure 2-8 shows a traditional linear graph for each example system. Note that both the RoboFlag and Jaybook examples define regions (rather than simple lines) because they can be operated in

flexible modes and those modes have different implications for dimensions such as how frequently the operator must intervene (Ifrq) and how much autonomy is afforded to the system (A).

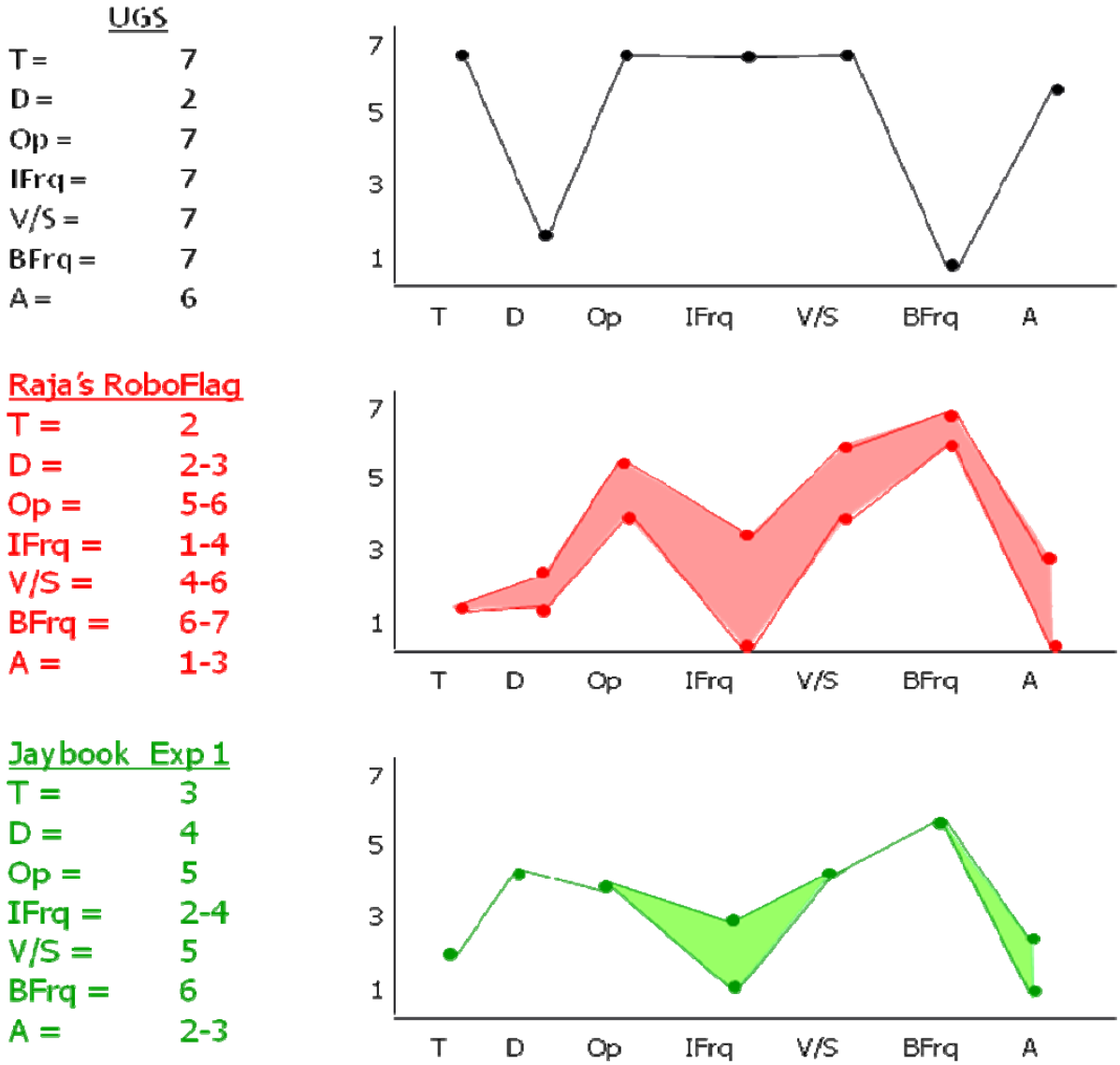


Figure 2-8: Linear Graphs of Values for Each of the Seven Dimensions for the Three Exemplar Systems.

These linear graphs make comparisons of the different examples quick and easy. It is obvious that Raja's RoboFlag and Jaybook are both capable of being operated in different modes, while UGSs are not. Furthermore, it is obvious that UGSs are operated in a much longer Typical mission (T) with many more "Vehicles" (VS) and with better (rarer) Intervention Frequency (IFrq), but with much less task Diversity (D) and Behavioral change Frequency (BFrq) than either of the others.

Perhaps more convenient and informative still, because we have structured the dimensional scales to be of equivalent lengths and orientations, we can graph them using a polar star format as in Figure 2-9. Furthermore, the polar star can group the dimensions in interesting categories, for example, task characteristics (Duration (T) and Diversity (D)) are shaded tan, operator characteristics (Operator Vehicle ratio (Op) and Intervention Frequency (IFrq)) are shaded pink, and platform/vehicle characteristics (Behavior Change Frequency (BFreq) and Vehicles/Sub-systems (VS)) are shaded green. Here, again, characteristic patterns are made visible. We can easily see that UGSs are very "good" for the operator in the sense that they require rare interventions and can support a high operator-to-vehicle ratio, but that they achieve this by severely restricting task diversity and behavior change frequency.

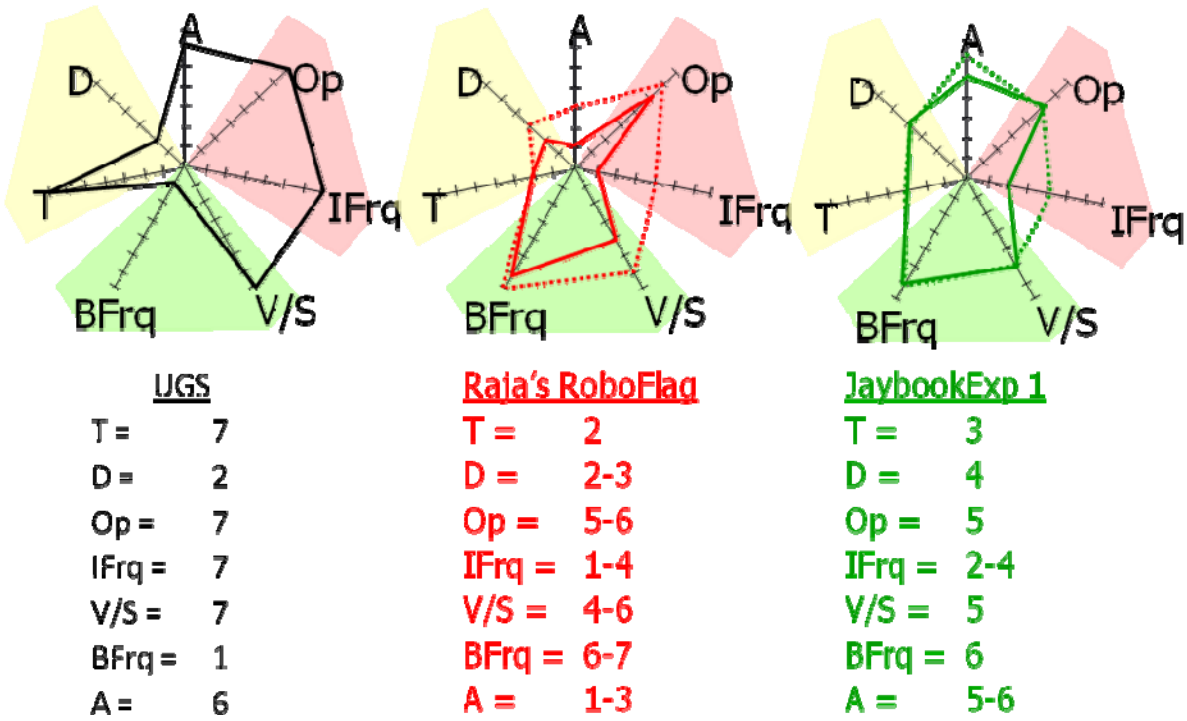


Figure 2-9: Polar Star Depictions of the Values for Each Example System. Note clustering of dimensions into Task characteristics (T and D in tan), Operator Characteristics (Op and IFrq in pink) and Platform or Vehicle Characteristics (BFrq and VS in green).

Strengths and Weaknesses

When initially presented, this 7D model was reasonably well received. It was seen as having the strengths of characterizing environment, vehicle, and operator characteristics, as well as the supervisory control relationship between them – and of doing so in relevant and interesting ways at a reasonable level of aggregation for the group. It was, however, also seen as still too complex for easy comprehension. Furthermore, there was a feeling that many of the dimensions were either poorly defined or confounded (non-orthogonal), or that the scales proposed were sub-optimal in some way. In practice, though, we began work on a simplified version of this seven dimensional model (as described next) to solve the complexity problem instead of concentrating on refining and improving the dimensions and scales.

2.7 2D INTERACTION DESCRIPTION

Given the feeling that even the 7D model described above was still too complex for our purposes, we sought to simplify it further. Upon thinking more deeply about the situation, we felt that supervisory control is, at its root, an interaction relationship between a supervisor and one or more subordinates. Everything else is external to that relationship and, while it is not incidental or unimportant, it certainly increases the complexity of a description. We could instead focus primarily and exclusively on the nature of that relationship. This insight led us to the two dimensional model of the supervisory control interaction and relationship described below. Dr. Miller first described and presented this model at the HFM-170 meeting in Paris in September of 2009, and expanded and provided examples of it with developed scales at the Dayton meeting in June of 2010.

After an analysis of various supervisory relationships in human-human interactions (see Figure 2-10), we concluded that they could be reasonably, and usefully, arrayed along two dimensions:

1) The attentional demand that the relationship required of the supervisory in order to accomplish any useful work; and
2) The "scope" or range of functions and capabilities that the subordinate(s) provide for performing useful work at that level of attentional demand.

	Demand	Scope	Useful?	Ratio
A Parking Valet	1	1	+	1
A Postal clerk/Fast Food Chef	1	2	+	.5
Shepherd to sheepdog	6	3	+	2
A young child (~3 yrs)	10	2	--	5
A teenager (~16 yrs)	8	6	-	1.3
A new secretary/asst	7	6	-	1.1
A concierge	3	4	+	.75
Quarterback (Am.) to team	2	4	+	.5
Project Manager to team	4	6	+	.67
CEO to corporation	3	8	+	.38
Presidential Aide	3	9	++	.33
Movie Director's Gopher	3	9	++	.33
Radar O'Reilly	1	10	++	.1

Figure 2-10: Scored Values for Attentional Demand and Performance Scope for Each of 13 Different Human-Human Supervisory Control Relationships.

At the time this initial thought exercise was performed, we had not developed scales for these dimensions, but we nevertheless took the step of informally rating each relationship on a 10 point scale ranging from low to high[3]. The results of this informal exercise are shown in Figure 2-10.

An interesting phenomena emerged from these ratings. It would seem that when the dimensions were arrayed against each other (see Figure 2-11), one could envision a "utility horizon" which ran roughly along the diagonal. "Useful" supervisory control relationships are those for which the cost to perform useful work is, at most, no more than the usefulness of the work performed. Since the attentional demand dimension we had identified was a fairly direct measure of the "cost" to the supervisor of performing the work and the "performance scope" term was a measure of the range of useful things the subordinate could perform, it served as at least an indirect measure for the benefit or usefulness of the work. Therefore, intuitively, relationships that fell on or below the diagonal were useful, while those which fell above the diagonal tended to be less useful. Such relationships could also be characterized somewhat more quantitatively by taking the ratio of the demand score to the scope score – as illustrated in the rightmost column of Figure 2-10. Here, higher values are indicative of less "useful" relationships (from the perspective of performing work with immediate utility), while lower values are indicative of more productive relationships. To check this intuition, we also provided intuitive ratings (on a five point scale from "- -" to "+ +") and then calculated a Pearson's correlation coefficient between the two sets (treating the second scale as -2 to +2). The resulting value was -.765, indicating that demand-to-scope ratio was highly negatively correlated with our sense of the usefulness of the relationship.

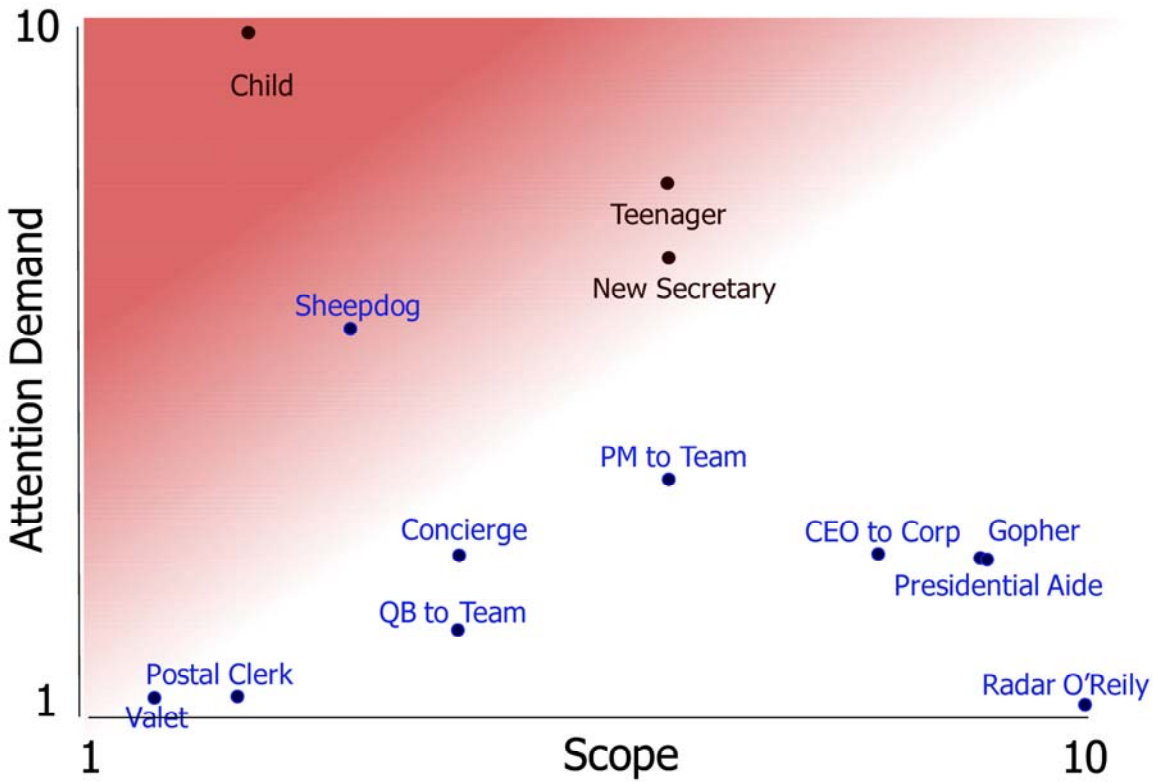


Figure 2-11: Graphing the Demand vs. Scope Scores to Illustrate a Diagonal "Utility Threshold".

[3] These ratings are Dr. Miller's alone.

The results of this initial brainstorming exercise were presented at the Paris meeting in 2009, and were met with general interest as a potentially promising direction. The primary criticism was that both the dimensions and the scales themselves were very informally defined. Thus, Dr. Miller took the step of trying to refine and quantify them, as described below.

Attentional Demand – This dimension was meant to capture the amount and frequency of time and attention required by the supervisor to manage the system and achieve the work desired. Our initial thought was that Olsen and Goodrich's [17] Fan Out metric was a close fit to what was needed. Olsen and Goodrich actually labeled their metric "Robot Attention Demand" (RAD) and defined it by the formula: IE / (IE + NT), where:

- IE is "Interaction Effort" – the time (or effort) required to interact with the robot; and
- NT is "Neglect Tolerance" – the robot's effective performance time without intervention.
- We presume (though this is **not** clearly stated in Olsen and Goodrich) that IE + NT = total time.

Thus, RAD is the proportion of time/effort during operation that the supervisor must devote to interacting with the automation. Based on RAD, we proposed "SAD" (System Attentional Demand) – the proportion of supervisor time/effort required to interact with the system in order to perform desired work. SAD is a unitless metric that ranges from 0 (for completely autonomous automation that requires no supervisory input) to 1 (no effectively "free" human time – the supervisory spends 100% of his/her time interacting to achieve the desired work).

To compare multiple systems with SAD, it is important to maintain consistent assumptions and scoring. Important considerations include:

- Whether/what "set up time" to include? The RAD definition was unclear (and, in fact, has been criticized by Crandall and Cummings [18]) for failing to consider pre-mission planning and configuration time, as well as engineering and design time. For using SAD or RAD to compare multiple systems, any set of practices with regards to these non-execution time parameters may be used, but it is important that they are applied consistently across systems that are analyzed.
- What performance context assumptions are used? Again, when assigning time and effort used to control and task the automation, it is also important to maintain consistency in assumptions across different systems rated if the goal is to compare those systems. For example, does the scenario of use represent a "sunny day" where nothing goes wrong or a worst case or factored error assumptions, etc.? Is the user considered a novice, an expert or somewhere in between? What error rates are assumed for the user's inputs?, etc. Again, the SAD metric can accommodate a wide range of different assumptions, but it is important that the assumptions be applied consistently across systems rated.

Performance Scope – There is a problem with using only SAD or RAD as the basis of comparison across supervisory control systems, however. Essentially, both compare systems *only* on the percentage of supervisor time they require; there is no explicit notion of the level of system effectiveness or work accomplished for a given level of SAD. In order to compare system functionality or effectiveness using SAD requires an assumption of homogenous tasks and performance targets – even Olsen and Goodrich's [18] Fan Out application of RAD presumed a homogenous task: "fanning out" a set of robots searching. That said, there is no explicit notion of the domain or task included in RAD or SAD. Using SAD alone, each of the following examples would have the same "attentional demand" value:

- Telling a fleet of 100 UAVs to "stay put" on the tarmac (that is, to do nothing);
- Telling very highly autonomous UAV to "execute" it's trip around the world; and

- Telling an efficient secretary to plan your next months' trips (assuming s/he already has access to your required trips and times and knowledge of your needs and preferences).

In each of these cases, the supervisor's attentional demand is one, short verbal interaction. On the other hand, the examples differ radically in the *scope of work performed*. Hence, we felt that a second dimension was needed to reflect the variety of tasks or functions the subordinate automation can perform. The problem is that tasks are inherently hierarchically decomposable and characterizing them across systems and domains is notoriously difficult. Therefore, in order to maintain some consistency in comparing different applications, we would need a *common* task model for the domain of interested which is shared by the applications/ relationships. This is not to say that all the systems must perform exactly the same tasks in the same way, but some basis for comparison across tasks was necessary – otherwise we would be stuck simply saying that the systems did different things. One way to accomplish this might be to require that the systems all accomplish a shared function or goal, though perhaps used different methods to do so.

Given such a model (which might, necessarily, be fairly abstract), we thought we could perhaps simply count the tasks (at a given level) that the proposed system performs, and that such a count would itself provide a metric for performance scope. The worked example to be described next was meant as a thought experiment to test whether a simple count of the tasks performed at a common level of a reference model could serve as a reasonable metric for performance scope.

An Elevator Example – To test this hypothesis, we conducted an extended thought experiment to compare several versions of a supervisor/subordinate system (which was, in most cases, also a human-automation system), each of which was designed to perform the same basic function: an elevator system in a multi-story building. A "reference model" for the tasks of elevator systems might be:

1) Summon/Initiate – call a/the elevator;
2) Select Elevator to respond;
3) Move to Called floor;
4) Control Speed;
5) Position Elevator Vertically;
6) Open Door;
7) Load Passenger(s);
8) Select Destination Floor;
9) Close Door;
10) Move to Destination Floor;
11) Control Speed;
12) Position Elevator Vertically;
13) Open Door;
14) Unload Passenger(s); and
15) Close Door.

Note that this is intended as a "spanning" model. Not all tasks are pertinent or performed by all systems, and not always in this order. The intent is that alternate elevator systems can be evaluated on whether and how they perform these tasks (with what mix of human and machine).

Next, we defined a set of human-elevator systems to map against the reference model drawn from the variety of elevator systems we had experienced:

1) *Old Style* – Completely Manual Operation, single elevator. In this style of elevator, the human (usually a dedicated "elevator operator") performed door opening/closing, vertical movement and positioning, etc.

2) *Freight Elevator* – Here, I was thinking of an elevator in an academic building at the University of Chicago where a button press controlled opening/closing the door but the human controlled the rest (positioning and movement, etc.).

3) *English "Moving Carriage"* – This was an elevator which I (and others) had experienced in England – where elevator "cars" ran on a continuous, non-stopping vertical track, there were no doors and riders stepped on/off the car as it passed by the opening on each floor.

4) *Current Single* – What we're all most familiar with: a modern "automated" elevator typical of moderate sized buildings. A button press summons the (single) elevator and which automatically opens its door when it arrives at the appropriate floor and then (usually) automatically closes the door when people board or leave. A different button is pressed for each floor desired and the elevator automatically travels to that floor, positions itself and opens its doors for riders to leave.

5) *Current Multiple* – What's in most big buildings, hotels: a bank of elevators for different floors/ regions. A single button is pressed to summon a car, but the automation behind the bank of elevators controls which elevator arrives at your floor. Riders enter and push buttons for their desired floor and the elevator automatically closes its doors and moves to the desired floor, where it opens its doors for disembarking.

6) *New York Marriott / HFES '08* – This was an advanced, optimized bank of elevators many of us encountered at the Marriott hotel in New York City at the Human Factors and Ergonomics Society meeting there in 2008. A user enters the desired floor in a central console and is told which elevator to go to. The elevator arrives, opens its doors and the user enters. There is no need to press a second button to indicate the desired floor, since this has already been done. Instead, the elevator moves to each of the floors users have indicated they want to go to – and supposedly does so somewhat more quickly since it is attempting to route users going to the same floors into a common car.

Given these example systems, we first attempted to determine a SAD metric for each of them. This was accomplished by estimating[4] the time required for each of the tasks in the reference model. The results are presented in Figure 2-12. Note that in order to provide a comparable number across the systems it was necessary to assume a common scenario. We chose a typical, shared task: going up 3 floors as a single passenger. Further, we noted that travel speed increases with more modern systems and, thus, total task time (IE + NT) decreases, tending to drive the SAD value higher than it would otherwise be[5]. In practice, getting there faster enables other work to be done by the human and, thus, perhaps we should have used the highest

[4] Note that, since this was a thought experiment, each of these estimates is based on the author's experience, memory, and judgment, not on empirically gathered data.

[5] A further, hypothetical system will illustrate this problem more dramatically: imagine a teleportation elevator system which requires that the user press a button to indicate which floor s/he wishes to go to and then *instantaneously* transports him/her there. Such a system would, in principle, require, say, 2 seconds for the user to press the initial button, but no additional time to get to the appropriate floor. Thus, IE would be 2s and IE+NT would also be 2s and SAD would be 2/2 = 1 – a value we associate with a fully manual system above. By contrast, if we took the IE time relative to the total time for the worst case, most manual comparable system (the "old style" elevator), we would have 2s / 247s = .008 – a very highly automated system. This seems to mesh with intuition more neatly.

value for total time across the systems which perform the comparable elevator function. This insight is not reflected in the values in Figure 2-12, however.

		Alternate Elevator Systems					
No.	Task	Old Style	Freight	Moving Carriage	Current Single	Current Multiple	Marriott/ HFES
1	Summon/Initiate	2s	2s	0s	2s	2s	2s
2	Select Elevator	0	0	0s	0s	0s	0s
3	Move to Called Floor	55s	55s	0s	0s	0s	0s
4	Control Speed	0s*	0s*	0s	0s	0s	0s
5	Position Vertically	15s	15s	0s	0s	0s	0s
6	Open Door	10s	2s	0s	0s	0s	0s
7	Load Passenger(s)	5s	5s	8s	5s	12s	8s
8	Select Destination Floor	--	--	3s	4s	2s	0s**
9	Close Door	10s	2s	0s	1s	1s	1s
10	Move to Destination	90s	90s	10s	0s	0s	0s
11	Control Speed	15s*	15s*	0s	0s	0s	0s
12	Position Vertically	15s	15s	0s	0s	0s	0s
13	Open Door	10s	2s	0s	0s	0s	0s
14	Unload Passenger(s)	5s	5s	8s	5s	5s	5s
15	Close Door	10s	2s	0s	0s	0s	0s
	TOTAL (IE)	242s	210s	29s	17s	22s	16s
	TOTAL TIME (IE + NT)	247s	247s	220s	150s	150s	150s
	SAD (IE/IE+NT)	.98	.85	.13	.11	.15	.11

* Extra effort beyond continuous commanded movement
** Combined with the summon/initiate act

Figure 2-12: SAD Estimates for 6 Different Elevator Systems.

We then created scope values for each of the alternate elevator systems by simply counting which of the tasks each mechanical (subordinate) system performed automatically. We quickly realized that many systems partially automated some of the tasks and we chose to use fractional values to indicate the degree to which, in the scorer's judgment, the system automated the task. The results are shown in Figure 2-13.

		Alternate Elevator Systems					
No.	Task	Old Style	Freight	Moving Carriage	Current Single	Current Multiple	Marriott/ HFES
1	Summon/Initiate	0	0	0	.3	.3	.3
2	Select Elevator	0	0	0	0	.7	1
3	Move to Called Floor	0	0	1	1	1	1
4	Control Speed	.5	.5	1	1	1	1
5	Position Vertically	0	0	.3	1	1	1
6	Open Door	0	.8	0	1	1	1
7	Load Passenger(s)	0	0	0	0	0	.3
8	Select Destination Floor	0	0	0	0	.1	.5
9	Close Door	0	.8	0	.8	.8	.8
10	Move to Destination	0	0	1	1	1	1
11	Control Speed	0	0	1	1	1	1
12	Position Vertically	0	0	.3	1	1	1
13	Open Door	0	.8	0	1	1	1
14	Unload Passenger(s)	0	0	0	0	.1	.3
15	Close Door	0	.8	0	.8	.8	.8
	TOTAL SCORE	.5	3.7	4.6	9.9	10.8	12
	Percent (out of 15)	3.3%	24.7%	30.7%	66%	72%	80%

Figure 2-13: Performance Scope Estimates for 6 Different Elevator Systems.

Armed with these sets of quantitative values, we were now able to, again, graph them in various ways to facilitate interpretation. Figure 2-14 shows the two dimensions plotted against each other. Interestingly, in this figure, those systems which are clearly less fully automated cluster in the upper left, while those which are more fully automated cluster in the lower right. This is in keeping with our intuitions that the more modern systems are, in fact, better representatives of “supervisory control” relationships while the older systems are poor examples of the relationship. This suggests that the diagonal in Figure 2-13 may represent a rough definitional boundary: those system which fall above it are not “supervisory control” systems precisely because they require too much effort from the human supervisor for the amount (scope) of work they accomplish. By contrast, those which fall below the line are good examples of supervisory control. The “moving carriage” example, which falls on the diagonal, is an interestingly ambiguous case. It automates some functions but still requires substantial vigilance and attention from the user and we are unsure whether to call it a supervisory control system or not.

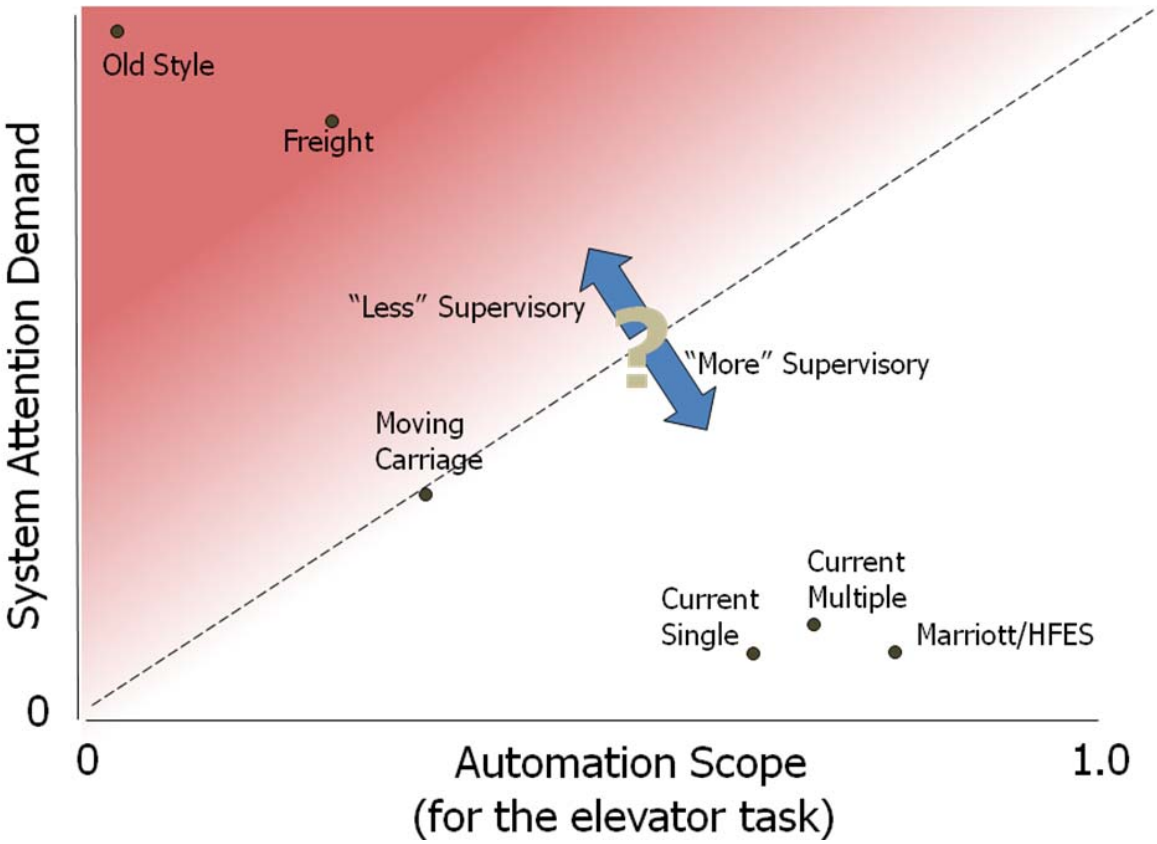


Figure 2-14: Graphing SAD and Performance Scope Dimensions Against Each Other – With a Suggestion of a Definitional Boundary for "Supervisory Control".

Figure 2-15 provides a slightly richer depiction of the graph by characterizing the different quadrants of it. Here, we might be able to assign labels to suggest the kinds of relationships which characterize the systems which fall into the various sections. For example, the upper left had quadrant is characterized by comparatively high attentional demand from the supervisor, but comparatively low scope of activities which the subordinate can perform. We might label relationships in this quadrant "child-like" since, like interacting with a child or infant, they require lots of supervision in order to perform little or no immediately useful work. Relationships falling into the upper right hand quadrant might be labeled "teenager-like" since, like interacting with a teenager, substantial supervision is still required, but a surprising range and scope of work can be accomplished if a supervisor is willing to take the time required. The lower left hand quadrant might be characterized as like interacting with a sheepdog since a sheepdog is capable of performing a limited range of behaviors, but can do so with very little supervision from the human supervisor. Finally, the lower right hand quadrant might be characterized as like an "Awesome Assistant" (e.g., a "Radar O'Riley" from the M*A*S*H television series) – someone who has a very wide range of performance capabilities and requires little supervision to perform them.

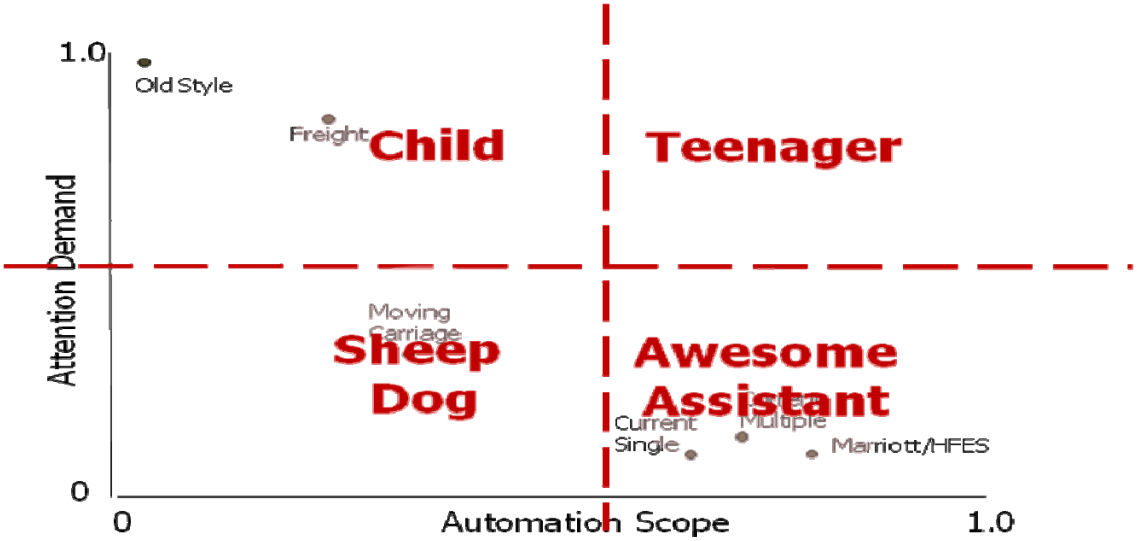


Figure 2-15: Labeling the Quadrants to Convey Intuitions About the Types of Relationships Afforded by Systems Which Fall Into Them.

Moving Beyond Elevators – The above thought experiment shows promise for this simplified 2D model since it illustrates the model's ability to capture interesting differences between a set of automation systems and to mirror our intuitions about their effectiveness and the degree to which they exemplify supervisory control relationships. Nevertheless, we realize that we have left open the question of whether or not this framework will prove relevant to real-world systems. Easily the most important challenge would be developing an acceptable "reference model" to evaluate performance scope for a set of realistic UMVs. While we did not perform this task, we were able to point to some characteristics of potential models to serve as starting points:

- It should characterize (and decompose) a common, shared function performed or goal achieved by all systems to be compared.
- Though the model of that shared function can be fairly abstract, it need be concrete enough to support deriving percentage time or effort estimates.
- It is helpful, but may not be required, if there are shared tasks in the decomposition of the shared function. If some systems require a sub-task to perform the function and others do not, the complete list for the reference model can include the union of all of the tasks and scope and SAD assessments can indicate whether or not the alternate systems perform the tasks and the time required.
- The reference model may need to be augmented by a specific, shared scenario (again, as performed by all systems to be compared) to enable temporal SAD computations.

While we did not develop such a model for UMV comparisons, one might be built out of shared vehicle functions such as navigation, propulsion, sensing, etc. One such model for aviation UMVs might be derived from typical functions of aircraft missions- such as those illustrated in the "automation trust" pyramid that Col. Jeff Eggers of the U.S. Air Force has created (see Figure 2-16 for Col. Eggers previously unpublished model). Col. Eggers uses this pyramid to convey the notion that trust in automation must be built from the bottom up, but it also serves as a general task or function decomposition for typical aircraft missions. It is likely that this model, or portions thereof, could serve as the basis for a reference model for at least UAVs for performing the type of SAD and Performance Scope analysis illustrated for elevators above.

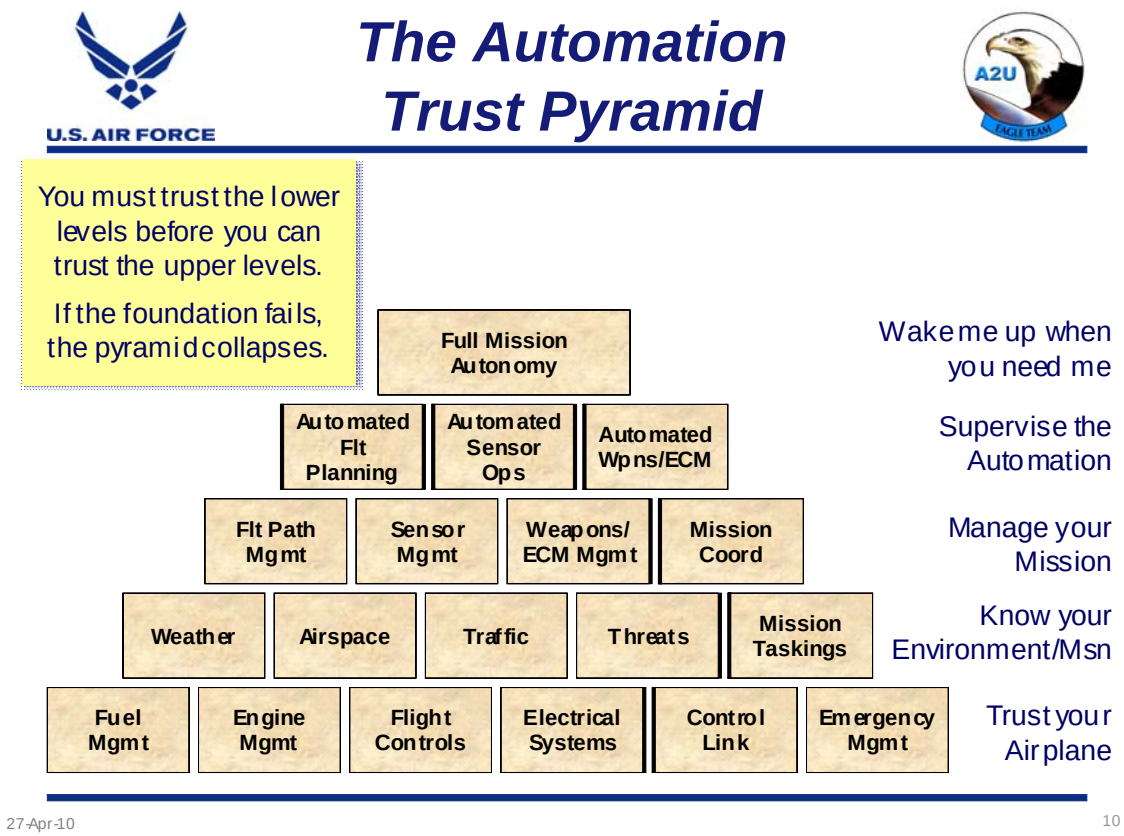


Figure 2-16: Col. Jeff Eggers' "Trust Pyramid" Which Represents a Typical Decomposition of Aviation Functions and Might be Useful as a Reference Model at Least for UAV Comparisons.

Strengths and Weaknesses

There was general consensus that this 2D model had done a reasonable job of operationalizing and quantifying the two dimensions and making them reusable across systems and applications to be analyzed. Similarly, this model has the strength of being very simple to explain and convey, thereby making it very suitable for use as an organizing framework for presenting the systems from this working group.

On the other hand, it was, perhaps, not quite as general as would be ideal due to the need for a shared reference model (which would necessarily be at least somewhat task and domain specific). Since we did not have time to complete investigating the development of a reference model for the supervisory control systems under investigation by the HFM-170 members, we cannot say with certainty whether a single, common reference model for all of our systems is possible. Some of us were, in fact, sceptical that a single reference model could encompass the air, sea, and ground applications being investigated, much less the component systems such as alternate visualizations or control systems to support supervisory control systems.

More seriously, though, there appeared to be general consensus that this 2D model may have gone too far in simplifying the characterization of supervisory control relationships, that it had suppressed too much interesting detail between the alternate systems. Having seen the results of this 2D model development, several group participants were interested in returning to (and further refining) the 7D model.

2.8 CONCLUSIONS

Clearly, this ongoing discussion of frameworks for supervisory control has illustrated that a very wide diversity of such models are possible, each with different strengths and weaknesses. While no single model emerged with which to present the results of this workshop, we did identify several dimensions that seem relevant to discriminating between supervisory control approaches being examined by this group, and we proposed methods for identifying and characterizing supervisory control relationships – particularly in the 2D model described above.

As has been noted before us (most notably by Parasuraman, Sheridan and Wickens, [6]), Sheridan's original model of levels of autonomy, while convenient, confounds many dimensions of supervisory control relationships and, ultimately, does not give a good sense of how such systems operate and what they do. Several alternate models have been proposed, including some in this document for the first time, which expand and refine our notion of supervisory control relationships as they exist in alternate systems. More importantly, these models have different strengths and weaknesses. Some are very detailed, specific and precise – but that very precision comes at a cost of both greater effort to construct representations of alternate systems within the framework and greater effort to understand the system characterization when it is later presented. Such approaches might be appropriate for design and evaluation of a given system (or, as with Miller and Parasuraman, [9], for conveying specific delegation actions to automation), but they are not particularly convenient for giving a "feel" of the system for comparison purposes. That said, any framework which does not express such precise details will, inevitably, suppress some aspects of system design or operation.

Our examination of the LoA^3 model (and, to a lesser degree, the 2D Interaction Description model) showed that, even though the term "supervisory control" arguably defines a relationship between supervisory and subordinate, any framework which concerns itself exclusively with this relationship and does not concurrently capture aspects of the operator, system and environment or task domain of usage is likely to be seen as insufficient. Instead, frameworks which seek to provide a basis for comparing and representing a set of alternate systems or approaches should also capture aspects of the equipment, personnel and context of usage – especially when those aspects vary in interesting ways from system to system.

Most of the models examined in the working group, and reported in this document, focused on the tasks or functions to be performed by the human + automation system. While there is an ongoing debate in the Human Factors community over the relative strengths and weaknesses of prescriptive task analysis vs. ecological function or goal analysis, the models proposed here are largely agnostic to the distinction. They are, however, focused on allocation of functions between human and automation in some fashion – whether by goal or state or function of scripted task. We believe this is due to the nature of supervisory control relationships – which were, after all, the focus of study. Supervisors necessarily retain some functions as their exclusive purview, share or retain others dynamically and in various combinations, and rely exclusively on their subordinates for performing still others.

At the end of this exercise, we believe that the 7D model held the most promise for satisfying the ends of this working group. This model was largely descriptive, but it captured several dimensions relevant to the alternate supervisory control systems, relationships and usages we were examining. While the specific dimensions examined might or might not be the best ones, and the scales for characterizing them might also be improved upon, this multi-dimensional description of alternate systems seemed to provide the right level and type of information for rapidly and easily conveying to ourselves and others how a set of supervisory control systems are similar and different from each other.

2.9 REFERENCES

[1] Sheridan, T. (2002). Humans and Automation: System Design and Research Issues. HFES/Wiley; Santa Monica, CA, USA.

[2] Box, G.E.P. and Draper, N. (1987). Empirical Model-Building and Response Surfaces, p. 424, Wiley.

[3] Sheridan, T. and Verplank, W. (1978). Human and computer control of undersea teleoperators (Tech. Rep.). Cambridge: Massachusetts Institute of Technology, Man-Machine Systems Laboratory.

[4] Miller, C. and Parasuraman, R. (2007). Designing for Flexible Interaction between Humans and Automation: Delegation Interfaces for Supervisory Control. Human Factors, 49(1), 57-75.

[5] Sheridan, T. (1987). Supervisory control. In G. Salvendy (Ed.), Handbook of human factors (pp. 1244-1268). New York: Wiley.

[6] Parasuraman, R., Sheridan, T. and Wickens, C. (2000). A model for types and levels of human interaction with automation. IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, 30, 286-297.

[7] Galster, S. (2003). An Examination of Complex Human-Machine System Performance Under Multiple Levels and Stages of Automation. Ph.D. Thesis. Catholic University.

[8] Diaper, D. (1989). Task analysis for human-computer interaction. Wiley; Chichester.

[9] Miller, C. and Parasuraman, R. (2003). Beyond Levels of Automation: An Architecture for More Flexible Human-Automation Collaboration. In Proceedings of the 47th Annual Meeting of the Human Factors and Ergonomics Society, October 13-17; Denver, CO, USA.

[10] Kirwan, B. and Ainsworth, L. (1993). A Guide to Task Analysis. Taylor and Francis; London.

[11] Parasuraman, R. and Miller, C.A. (2006). Delegation Interfaces for Human Supervision of Multiple Unmanned Vehicles: Theory, Experiments, and Practical Applications. In Cooke, N. Pringle, H., Pedersen, H., Conner, O. (Eds.) Human Factors of Remotely Piloted Vehicles. Elsevier; Amsterdam.

[12] Parasuraman, R., Galster, S., Squire, P., Furukawa, H. and Miller, C. (2005). A Flexible Delegation-Type Interface Enhances System Performance in Human Supervision of Multiple Robots: Empirical Studies with RoboFlag. *IEEE Systems, Man and Cybernetics – Part A, Special Issue on Human-Robot Interactions, 35*(4), 481-493.

[13] Squire, P., Furukawa, H., Galster, S., Miller, C. and Parasuraman, R. (2004) Adaptability and flexibility are key! Benefits of the "Playbook" interface for human supervision of multiple unmanned vehicles. In Proceedings of the 2004 Meeting of the Human Factors and Ergonomics Society, September 20-24; New Orleans, LA, USA.

[14] Parasuraman, R., Galster, S. and Miller, C. (2003). Human Control of Multiple Robots in the RoboFlag Simulation Environment. In Proceedings of the 2003 Meeting of the IEEE Systems, Man and Cybernetics society. October 5-8; Washington, DC, USA.

[15] Fern, L. and Shively, R.J. (2009). "A comparison of varying levels of automation on the supervisory control of multiple UASs", Proc. AUVSI North America, Washington, DC, USA.

[16] Flaherty, S.R. and Shively, R.J. (2010). Delegation control of multiple unmanned systems. Proceedings of SPIE Defense, Security, and Sensing 2010. Orlando, FL, USA.

[17] Olsen, D. and Goodrich, M. (2003). Metrics for evaluating human-robot interactions. In Proc. NIST Performance Metrics for Intelligent Systems Workshop.

[18] Crandall, J. and Cummings, M. (2007). Developing performance metrics for the supervisory control of multiple robots. In Proceedings of the 2007 ACM/IEEE Conference on Human-Robot Interaction (HRI'07), 33-40.