26. Collaborative Human–Automation Decision Making

Mary L. Cummings, Sylvain Bruni

The development of a comprehensive collaborative human–computer decision-making model is needed that demonstrates not only what decision-making functions should or could be assigned to humans or computers, but how many functions can best be served in a mutually supportive environment in which the human and computer collaborate to arrive at a solution superior to that which either would have come to independently. To this end, we present the human–automation collaboration taxonomy (HACT), which builds on previous research by expanding the *Parasuraman* information processing model [26.1], specifically the decision-making component. Instead of defining a simple level of automation for decision making, we deconstruct the process to include three distinct roles: the moderator, generator, and decider. We propose five levels of collaboration (LOCs) for each of these roles, which form a three-tuple that can be analyzed to evaluate system collaboration, and possibly identify areas for design intervention. A resource allocation mission planning case study is presented using this framework to illustrate the benefit for system designers.

In developing any complex supervisory control system that involves the integration of human decision making with automation, the question often arises as to where, how, and how much humans should be in the decision-making loop. Allocating roles and functions between the human and the computer is critical in defining efficient and effective system architectures. However, role allocation does not necessarily need to be mutually exclusive, and instead of systems that clearly define specific roles for either human or automation, it is possible that humans and computers can collaborate in a mutually supportive decision-making environment. This is especially true for aspects of supervisory control that include planning and resource allocation (e.g., how should multiple aircraft be routed to avoid bad weather, or how to allocate ambulances in a disaster), which is the focus of this chapter. For discussion purposes, we define collaboration as *the mutual engagement of agents in a coordinated and synchronous effort to solve a problem based on a shared conception of it* [26.2, 3]. We define agents as either humans or some form of automation/computer that provides some level of interaction.

For planning and resource allocation supervisory control tasks in complex systems, the problem spaces are large with significant uncertainty, so the use of automation is clearly warranted in attempting to solve a particular problem; for example, if bad weather prevents multiple aircraft from landing at an airport, air-traffic controllers need to know right away which alternate airports are within fuel range, and of these, which have the ability to service the different aircraft types, the predicted traffic volume, routing conflicts, etc. While automation could be used to provide optimized routing recommendations quickly, computer-generated solutions are unfortunately not always the best solutions. While fast and able to handle complex computation far better than humans, computer optimization algorithms are notoriously *brittle* in that they can only take into account those quantifiable variables identified in the design stages that were deemed to be critical [26.4]. In supervisory control systems with inherent uncertainties (weather impacts, enemy movement, etc.), it is not possible to include a priori every single variable that could impact the final solution. Moreover, it is not clear exactly what characterizes an *optimal* solution in uncertain such scenarios. Often, in these domains, the need to generate an *optimal* solution should be weighed against a *satisficing* [26.5] solution. Because constraints and variables are often dynamic in complex supervisory control environments, the definition of optimal is also a constantly changing concept. In those cases of time pressure, having a solution that is good enough, robust, and quickly reached is often preferable to one that requires complex computation and extended periods of times, which may not be accurate due to incorrect assumptions.

Recognizing the need for automation to help navigate complex and large supervisory control problem spaces, it is equally important to recognize the critical role that humans play in these decision-making tasks. Optimization is a word typically associated with computers but humans are natural optimizers as well, although not necessarily in the same linear vein as computers. Because humans can reason inductively and generate conceptual representations based on both abstract and factual information, they also have the ability to optimize based on qualitative and quantitative information [26.6]. In addition, allowing operators active participation in decision-making processes provides not only safety benefits, but promotes situation awareness and also allows a human operator, and thus a system, to respond more flexibly to uncertain and unexpected events. Thus, decision support systems that leverage the collaborative strength of humans and automation in supervisory control planning and resource allocation tasks could provide substantial benefits, both in terms of human and system performance,

Unfortunately, little formal guidance exists to aid designers and engineers in the development of collaborative human–computer decision support systems. While many frameworks have been proposed that detail levels of human–automation role allocation, there has been no focus on what specifically constitutes collaboration in terms of role allocation and how this can be quantified to allow for specific system analysis as well as design guidance. Therefore, to better describe human-collaborative decision support systems in order to provide more detailed design guidance, we present the human–automation collaboration taxonomy (HACT) [26.7].

26.1 Background

There is little previous literature that attempts to classify, describe, or provide design guidance on human–automation (or computer) collaboration. Most previous efforts have generally focused on developing application-specific decision support tools that promote some open-ended form of human–computer interaction (e.g., [26.8–10]). In an attempt to categorize human–computer collaboration more formally, *Silverman* [26.11] proposed categories of human–computer interaction in terms of critiquing, although this is a relatively narrow field of human–computer collaboration. *Terveen* [26.12] attempted to seek some unified approach and more broadly define and categorize human–computer collaboration in terms of human emulation and human "complementary" [sic]. Beyond these broad definitions and categorizations of human–computer collaboration and narrow applications of specific algorithms and visualizations, there has been no underlying theory addressing how collaboration with an automated agent supports operator decision making at the most fundamental information processing level.

So while the literature on human–automation *collaboration* in decision making is sparse, the converse is true in terms of scales and taxonomies of automation levels that describe *interactions* between a human operator and a computer/automation. These levels of automation (LOAs) generally refer to the role allocation between automation and the human, particularly in the analysis and decision phases of a simplified information processing model of acquisition, analysis, decision, and action phases [26.1, 13, 14]. The originators of the concept of levels of automation, *Sheridan* and *Verplank* (SV), initially proposed that automation could range from a fully manual system with no computer intervention to a fully automated system where the human is kept completely out of the loop [26.15]. *Parasuraman* [26.1] expanded the original SV LOA to include ten levels (Table 26.1).

Table 26.1 Levels of automation (after [26.1, 15])

Automation level	Automation description
1	The computer offers no assistance: human must take all decision and actions
2	The computer offers a complete set of decision/action alternatives, or
3	Narrows the selection down to a few, or
4	Suggests one alternative, and
5	Executes that suggestion if the human approves, or
6	Allows the human a restricted time to veto before automatic execution, or
7	Executes automatically, then necessarily informs humans, and
8	Informs the human only if asked, or
9	Informs the human only if it, the computer, decides to
10	The computer decides everything and acts autonomously, ignoring the human

At the lower levels, LOAs 1–4, the human is actively involved in the decision-making process. At level 5, the automation takes on a more active role in executing decisions, while still requiring consent from the operator before doing so (known as management-by-consent). Level 6, typically referred to as management-by-exception, allows the automation a more active role in decisions, executing solutions unless vetoed by the human. For levels 7–10, humans are only allowed to accept or veto solutions presented to them. Thus, as levels increase, the human is increasingly removed from the decision-making loop, and the automation is increasingly allocated additional authority. This scale addresses primarily authority allocation, i. e., who is given the authority to make the final decision, although only to a much smaller and limited degree does it address the solution-generation aspect of decision making, which is a critical aspect of human–computer collaboration.

The solution-generation process in supervisory control planning and resource allocation tasks is critical because this is the aspect of the human–computer interaction where the variables and constraints can be manipulated to determine solution alternatives. This access creates a sensitivity analysis trade space that allows human operators the ability to cope with uncertainty and apply judgment and experience that are unavailable to computer algorithms. While the LOAs in Table 26.1 provide some indirect guidance as to how the solution-generation process can be allocated either to the human or computer, it is only tangentially inferred, and there is no level that allows for joint construction or modification of solutions.

Other LOA taxonomies have addressed the need to examine authority and solution generation LOAs, although none have addressed them in an integrated fashion; for example, *Endsley* [26.16] incorporated artificial intelligence into a five-point LOA scale, thus addressing some aspects of solution generation and authority. *Riley* [26.17] investigated the use of the level of information attribute in addition to the automation authority attribute, creating a two-dimensional scale. Another ten-point scale was created by *Endsley* and *Kaber* [26.16] where each level corresponds to a specific task behavior of the automation, going from *manual control* to *full automation*, through intermediate levels such as *blended decision making* or *supervisory control*. While all of these scales acknowledge that there are possible collaborative processes between humans and automated agents, none specifically detail how this interaction can occur, and how different attributes of a collaborative system can each have a different LOA. To address this shortcoming in the literature, we developed the human–automation collaboration taxonomy (HACT), which is detailed in the next section.

26.2 The Human–Automation Collaboration Taxonomy (HACT)

In order to better understand how human operators and automation collaborate, the four-stage information-processing flow diagram of *Parasuraman* [26.1] (with stages: information acquisition, information analysis,

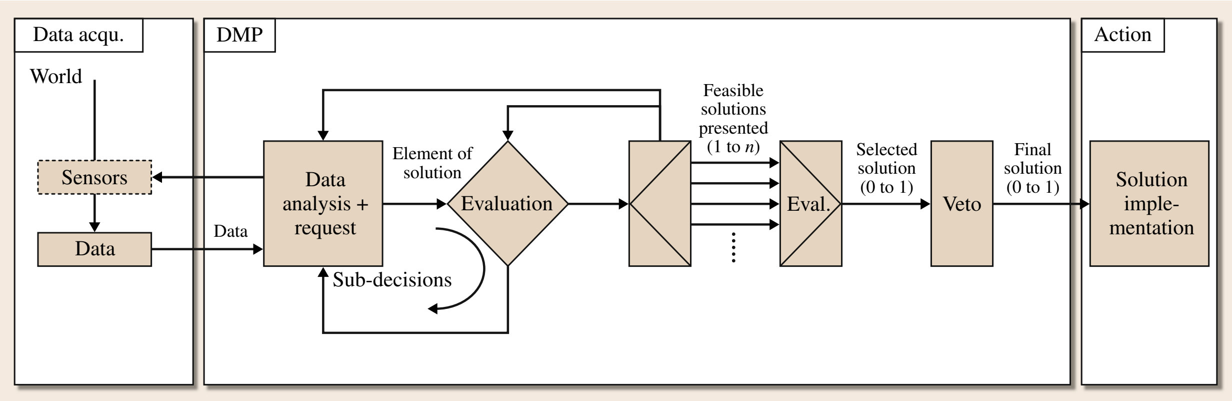


Fig. 26.1 The HACT collaborative information-processing model

decision selection, and action implementation) was modified to focus specifically on collaborative decision making. This new model, shown in Fig. 26.1, features three steps: data acquisition, decision making, and action taking. The data acquisition step is similar to that proposed by *Parasuraman* [26.1] in that sensors retrieve information from the outside world or environment, and transform it into working data. The collaborative aspect of this model occurs in the next stage, the decision-making process, which corresponds to the integration of the analysis and decision phases of the *Parasuraman* [26.1] model.

First, the data from the acquisition step is analyzed, possibly in an iterative way where requests for more data can be sent to the sensors. The data analysis outputs some elements of a solution to the problem at hand. The evaluation block estimates the appropriateness of these elements of solutions for a potential final solution. This block may initiate a recursive loop with the data analysis block; for instance, operators may request more analysis of the domain space or part thereof. At this level, subdecisions are made to orient the search and analysis process. Once the evaluation step is validated, i. e., subdecisions are made, the results are assembled to constitute one or more feasible solutions to the problem. In order to generate feasible solutions, it is possible to loop back to the previous evaluation phase, or even to the data analysis step. At some point, one or more feasible solutions are presented in a second evaluation step.

The operator or automation (depending on the level of automation) will then select one solution (or none) out of the pool of feasible solutions. After this selection procedure, a veto step is added, since it is possible for one or more of the collaborating agents to veto the solution selected (such as in management-by-exception). An agent may be a human operator or an automated computer system, also called automation. If the proposed solution is vetoed, the output of the veto step is empty, and the decision-making process starts again. If the selected solution is not vetoed, it is considered the *final solution* and is transferred to the action mechanism for implementation.

26.2.1 Three Basic Roles

Given the decision-making process (DMP) shown in Fig. 26.1, three key roles have been identified: moderator, generator, and decider. In the context of collaborative human–computer decision making, these three roles are fulfilled either by the human operator, by automation, or by a combination of both. Figure 26.2 displays how these three basic roles fit into the HACT collaborative information-processing model. The generator and the decider roles are mutually exclusive in that the domain of competency of the generator (as outlined in Fig. 26.2) does not overlap with that of the decider. However, the moderator's role subsumes the entire decision-making process. As will be discussed, each of the three roles has its own possible LOA scale.

The Moderator

The moderator is the agent(s) that keeps the decision-making process moving forward, and ensures that the various phases are executed; for instance, the moderator may initiate the decision-making process and interaction between the human and automation. The moderator may prompt or suggest that subdecisions need to be made, or evaluations need to be considered. It could also be involved keeping the decision processing within

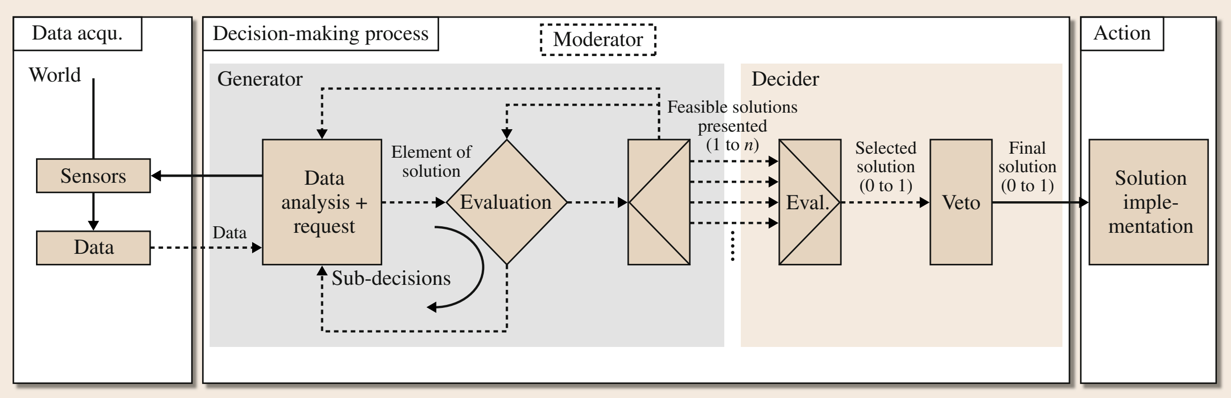


Fig. 26.2 The three collaborative decision-making process roles: moderator, generator, and decider

prespecified limits when time pressure is a concern. In relation to the ten-level SV LOA scale (Table 26.1), the step between LOA 4 and 5 implies this role, but does not address the fact that moderation can occur across multiple segments of the decision-making process and separate from the tasks of solution generation and selection.

The Generator

The generator is the agent(s) that generates feasible solutions from the data. Typically, the generator role involves searching, identifying, and creating solution(s) or parts thereof. Most of the previously discussed LOAs (e.g., [26.1, 16]) address the role of a solution generator. However, instead of focusing on only the actual solution (e.g., automation generating one or many solutions), we expand in detail the notion of the generator to include other aspects of solution generation, i. e., all the other steps within the generator box (Fig. 26.2), such as the automation analyzing data, which makes the solution generation easier for the human operator. Additionally, the role allocation for generator may not be mutually exclusive but could be shared to varying degrees between the human operator and the automation; for example, in one system the human could define multiple constraints and the automation searches for a set of possible solutions bounded by these constraints. In another system, the automation could propose a set of possible solutions and then the human operator narrows down these solutions.

Table 26.2 Moderator and generator levels

Level	Who assumes the role of generator and/or moderator?
2	Human
1	Mixed, but more human
0	Equally shared
−1	Mixed, but more automation
−2	Automation

For both the moderator and generator roles, the general LOAs can be seen in Table 26.2, which we recharacterize as LOCs (levels of collaboration). While the levels could be parsed into more specific levels, as seen in previously discussed LOAs, these five levels were chosen to reflect degrees of collaboration with the center scale reflecting balanced collaboration. At either end of the LOC scale (2 or −2), the system, in terms of moderation and generation, is not collaborative. The negative sign should not be interpreted as a critical reflection on the use of automation; it simply reflects scaling in the opposite direction. A system at LOC 0, however, is a balanced collaborative system for either the moderator and/or generator.

The Decider

The third role within the HACT collaborative decision-making process is the decider. The decider is the agent(s) that *makes the final decision*, i. e., that selects the potentially final solution out of the set of feasible solutions presented by the generator, and who has veto power over this selection decision. Veto power is a nonnegotiable attribute: once an agent vetoes a decision, the other agent cannot supersede it. This veto power is also an important attribute in other LOA scales [26.1, 16], but we have added more resolution to the possible role allocations in keeping with our collaborative approach, listed in Table 26.3. As in Table 26.2, the most balanced

Table 26.3 Decider levels

Level	Who assumes the role of decider?
2	Human makes final decision, automation cannot veto
1	Human or automation can make final decision, human can veto, automation cannot veto
0	Human or automation can make final decision, human can veto, automation can veto
−1	Human or automation can make final decision, human cannot veto, automation can veto
−2	Automation makes final decision, human cannot veto

collaboration between the human and the automation is seen at the midpoint, with the greatest lack of collaboration at the extreme levels.

The three roles, moderator, generator and decider, focus on the tasks or actions that are undertaken by the human operator, the automation, or the combination of both within the collaborative decision-making process.

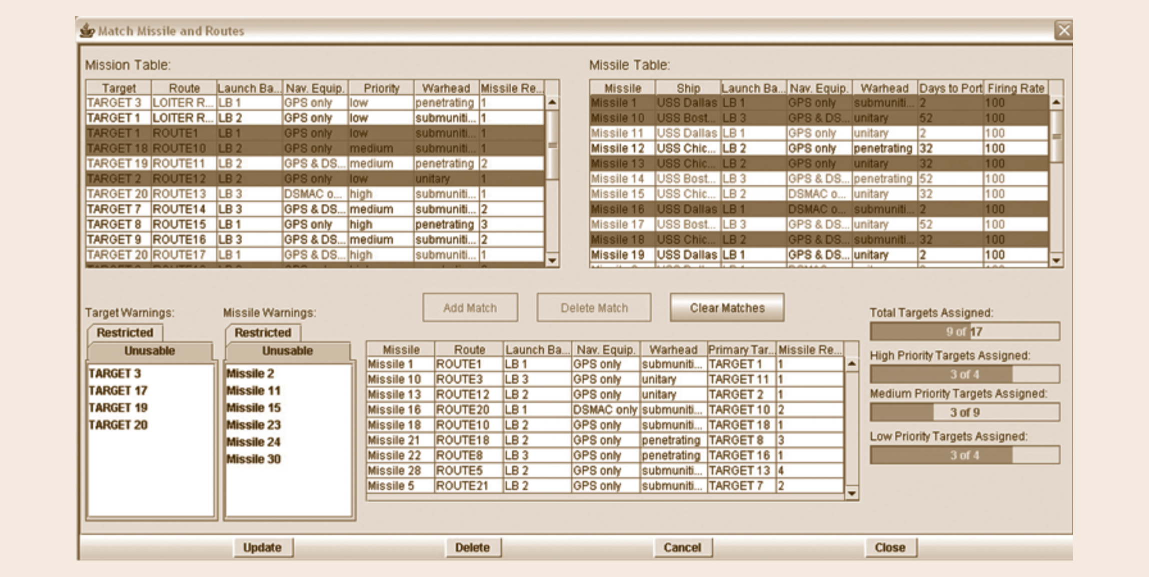
26.2.2 Characterizing Human Supervisory Control System Collaboration

Given the scales outlined above, decision support systems can be categorized by the collaboration across the three different roles (moderator, generator, and decider) in the form of a three-tuple, e.g., (2, 1, 2) or (−2, −2, 1). In the first example of (2, 1, 2), this system includes the human as both the moderator and the decider, as well as generating most of the solution, but leverages some automation for the solution generation. An example of such a system would be one where an operator needs to plan a mission route but must select not just the start and goal state, but all intermediate points in order to avoid all restricted zones and possible hazards. Automation is used to ensure fuel limits are not exceeded and to alert the operator in the case of any area violations.

This is in contrast to the highly automated (−2, −2, 1) example, which is the characterization of the Patriot missile system. This antimissile missile system notifies the operator that a target has been detected, allows the operator approximately 15 s to veto the automation's solution, and then fires if the human does not intervene. Thus the automation moderates the flow, analyzes the solution space, presents a single solution, and then allows the human to veto this. Note that under the ten LOAs in Table 26.1, this system would be characterized at LOA 6, but the HACT three-tuple provides much more information. It demonstrates that the system is highly automated at the moderator and generator levels, while the human has more authority than the automation for the final decision. However, a low decider level does not guarantee a human-centered system in that the Patriot system has accidentally killed three North Atlantic Treaty Organization (NATO) airmen because operators were not able to determine in the 15 s window that the targets were actually friendly aircraft and not enemy missiles. This example illustrates that all three entries in the HACT taxonomy are important for understanding a system's collaborative potential.

26.3 HACT Application and Guidelines

In order to illustrate the application and utility of HACT, a case study is presented. Given the increased complexity, uncertainty, and time pressure of mission planning and resource allocation in command and control settings, increased automation is an obvious choice for system improvement. However just what level of automation/collaboration should be used in such an application is not so obvious. As previously mentioned, too much automation can induce complacency and loss of situation awareness, and coupled with the inherent inability of automated algorithms to be perfectly correct in dynamic command and control settings, high levels of automation are not advisable. However, low levels of automation can cause unacceptable operator workload as well as suboptimal, very inefficient solutions. Thus the resource allocation aspect of mission planning



Mission Table:

Target	Route	Launch Ba...	Nav. Equip.	Priority	Warhead	Missile Re...
TARGET 3	LOITER R...	LB 1	GPS only	low	penetrating	1
TARGET 1	LOITER R...	LB 2	GPS only	low	submuniti...	1
TARGET 1	ROUTE1	LB 1	GPS only	low	submuniti...	1
TARGET 18	ROUTE10	LB 2	GPS only	medium	submuniti...	1
TARGET 19	ROUTE11	LB 2	GPS & DS...	medium	penetrating	2
TARGET 2	ROUTE12	LB 2	GPS only	low	unitary	1
TARGET 20	ROUTE13	LB 3	DSMAC o...	high	submuniti...	1
TARGET 7	ROUTE14	LB 3	GPS & DS...	medium	submuniti...	2
TARGET 8	ROUTE15	LB 1	GPS only	high	penetrating	3
TARGET 9	ROUTE16	LB 3	GPS & DS...	medium	submuniti...	2
TARGET 20	ROUTE17	LB 1	GPS & DS...	high	submuniti...	1

Missile Table:

Missile	Ship	Launch Ba...	Nav. Equip.	Warhead	Days to Port	Firing Rate
Missile 1	USS Dallas	LB 1	GPS only	submuniti...	2	100
Missile 10	USS Bost...	LB 3	GPS & DS...	unitary	52	100
Missile 11	USS Dallas	LB 1	GPS only	unitary	2	100
Missile 12	USS Chic...	LB 2	GPS only	penetrating	32	100
Missile 13	USS Chic...	LB 2	GPS only	unitary	32	100
Missile 14	USS Bost...	LB 3	GPS & DS...	penetrating	52	100
Missile 15	USS Chic...	LB 2	DSMAC o...	unitary	32	100
Missile 16	USS Dallas	LB 1	DSMAC o...	submuniti...	2	100
Missile 17	USS Bost...	LB 3	GPS & DS...	unitary	52	100
Missile 18	USS Chic...	LB 2	GPS & DS...	submuniti...	32	100
Missile 19	USS Dallas	LB 1	GPS & DS...	unitary	2	100

Target Warnings:

TARGET 3
TARGET 17
TARGET 19
TARGET 20

Missile Warnings:

Missile 2
Missile 11
Missile 15
Missile 23
Missile 24
Missile 30

Missile	Route	Launch Ba...	Nav. Equip.	Warhead	Primary Tar...	Missile Re...
Missile 1	ROUTE1	LB 1	GPS only	submuniti...	TARGET 1	1
Missile 10	ROUTE3	LB 3	GPS only	unitary	TARGET 11	1
Missile 13	ROUTE12	LB 2	GPS only	unitary	TARGET 2	1
Missile 16	ROUTE20	LB 1	DSMAC only	submuniti...	TARGET 10	2
Missile 18	ROUTE10	LB 2	GPS only	submuniti...	TARGET 18	1
Missile 21	ROUTE18	LB 2	GPS only	penetrating	TARGET 8	3
Missile 22	ROUTE8	LB 3	GPS only	penetrating	TARGET 16	1
Missile 28	ROUTE5	LB 2	GPS only	submuniti...	TARGET 13	4
Missile 5	ROUTE21	LB 2	GPS only	submuniti...	TARGET 7	2

Total Targets Assigned: 9 of 17

High Priority Targets Assigned: 3 of 4

Medium Priority Targets Assigned: 3 of 9

Low Priority Targets Assigned: 3 of 4

Fig. 26.3 Interface 1

is well suited for some kind of collaborative human–computer environment. To investigate this issue, three interfaces were designed for a representative system, each with a different LOA/LOC detailed in the next section.

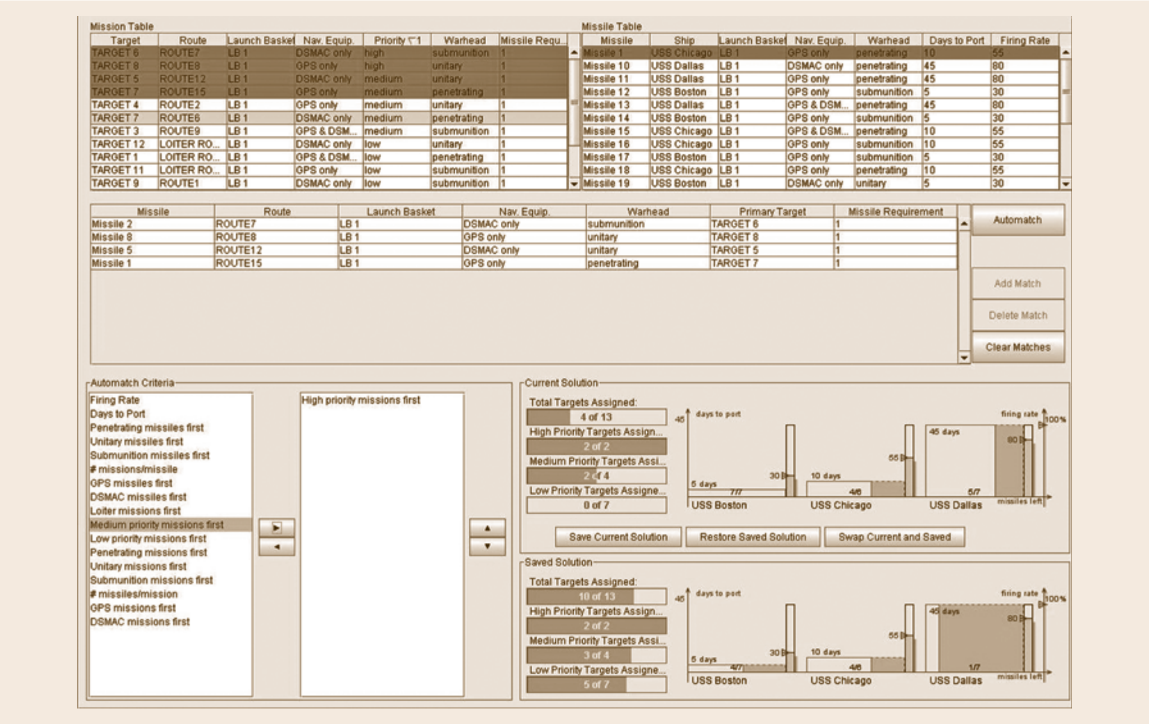
The general objective of this resource allocation problem is for an operator to match a set of military missions with a set of available resources, in this case Tomahawk missiles aboard ship and submarine launch platforms.

Interface 1 (Fig. 26.3) was designed to support manual matching of the missiles to the missions at a low level of collaboration. This interface provides raw data tables with all the characteristics of missions and missiles that must be matched, but only provides very limited automated support, such as basic data sorting, mission/missile assignment summaries by categories, and feedback on mission–missile incompatibility and current assignment status. Therefore, this interface mostly involves manual problem solving. As a result, interface 1 is assigned a level 2 moderator because the human operator fully controls the process. Because interface 1 only features basic automation support, the generator role is at level 1. The decider is at level 2 since only the human operator can validate a solution for further implementation, with no possible automation veto.

Interface 2 (Fig. 26.4) was designed to offer the human operator the choice to either solve the mission–missile assignment task manually as in interface 1 (note in Fig. 26.4 that the top part of interface 2 is a replica of interface 1 shown in Fig. 26.3), or to leverage automation and collaborate with the computer to generate solutions. In the latter instance, termed Automatch, the human operator can steer the search of the automated solution in the domain space by selecting and prioritizing search criteria. Then, the automation's fast computing capabilities perform a heuristic search based on the criteria defined by the human. The operator can either keep the solution output or modify it manually. The operator can also elect to modify the search criteria to get a new solution.

Therefore, for interface 2, the moderator remains at level 2 because the human operator is still in full control of the process, including which tasks are completed, at what pace, and in which order. Because of the flexibility in obtaining a solution in that the human can define the search criteria, thus orienting the automation which does the bulk of the computation, the generator is labeled 0. The decider is at level 2 since only the human operator can validate a final solution, which the automation cannot veto.

While interfaces 1 and 2 are both based on the use of raw data, interface 3 (Fig. 26.5) is completely graphical, and allows the operator to only have access to postsolution sensitivity analysis tools. For interface 2, the automated solution process is guided by the human, who also can conduct sensitivity analysis via an Au-



Mission Table

Target	Route	Launch Basket	Nav. Equip.	Priority ▽1	Warhead	Missile Requ...
TARGET 6	ROUTE7	LB 1	DSMAC only	high	submunition	1
TARGET 8	ROUTE8	LB 1	GPS only	high	unitary	1
TARGET 5	ROUTE12	LB 1	DSMAC only	medium	unitary	1
TARGET 7	ROUTE15	LB 1	GPS only	medium	penetrating	1
TARGET 4	ROUTE2	LB 1	GPS only	medium	unitary	1
TARGET 7	ROUTE6	LB 1	DSMAC only	medium	penetrating	1
TARGET 3	ROUTE9	LB 1	GPS & DSM...	medium	submunition	1
TARGET 12	LOITER RO...	LB 1	DSMAC only	low	unitary	1
TARGET 1	LOITER RO...	LB 1	GPS & DSM...	low	penetrating	1
TARGET 11	LOITER RO...	LB 1	GPS only	low	submunition	1
TARGET 9	ROUTE1	LB 1	DSMAC only	low	submunition	1

Missile Table

Missile	Ship	Launch Basket	Nav. Equip.	Warhead	Days to Port	Firing Rate
Missile 1	USS Chicago	LB 1	GPS only	penetrating	10	55
Missile 10	USS Dallas	LB 1	DSMAC only	penetrating	45	80
Missile 11	USS Dallas	LB 1	GPS only	penetrating	45	80
Missile 12	USS Boston	LB 1	GPS only	submunition	5	30
Missile 13	USS Dallas	LB 1	GPS & DSM...	penetrating	45	80
Missile 14	USS Boston	LB 1	GPS only	submunition	5	30
Missile 15	USS Chicago	LB 1	GPS & DSM...	penetrating	10	55
Missile 16	USS Chicago	LB 1	GPS only	submunition	10	55
Missile 17	USS Boston	LB 1	GPS only	submunition	5	30
Missile 18	USS Chicago	LB 1	GPS only	penetrating	10	55
Missile 19	USS Boston	LB 1	DSMAC only	unitary	5	30

Missile	Route	Launch Basket	Nav. Equip.	Warhead	Primary Target	Missile Requirement
Missile 2	ROUTE7	LB 1	DSMAC only	submunition	TARGET 6	1
Missile 8	ROUTE8	LB 1	GPS only	unitary	TARGET 8	1
Missile 5	ROUTE12	LB 1	DSMAC only	unitary	TARGET 5	1
Missile 1	ROUTE15	LB 1	GPS only	penetrating	TARGET 7	1

Automatch Criteria

Firing Rate
Days to Port
Penetrating missiles first
Unitary missiles first
Submunition missiles first
# missions/missile
GPS missiles first
DSMAC missiles first
Loiter missions first
Medium priority missions first
Low priority missions first
Penetrating missions first
Unitary missions first
Submunition missions first
# missiles/mission
GPS missions first
DSMAC missions first

High priority missions first

Current Solution

Total Targets Assigned: 4 of 13
High Priority Targets Assign... 2 of 2
Medium Priority Targets Assi... 2 of 4
Low Priority Targets Assigne... 0 of 7

Saved Solution

Total Targets Assigned: 10 of 13
High Priority Targets Assign... 2 of 2
Medium Priority Targets Assi... 3 of 4
Low Priority Targets Assigne... 5 of 7

Fig. 26.4 Interface 2

tomatch function; the Automatch button at the top of interface 3 is similar to that in interface 2. However, the user can only select a limited subset of information criteria by which to orient the algorithmic search, causing the operator to rely more on the automation than in interface 2. Thus the HACT three-tuple in this case

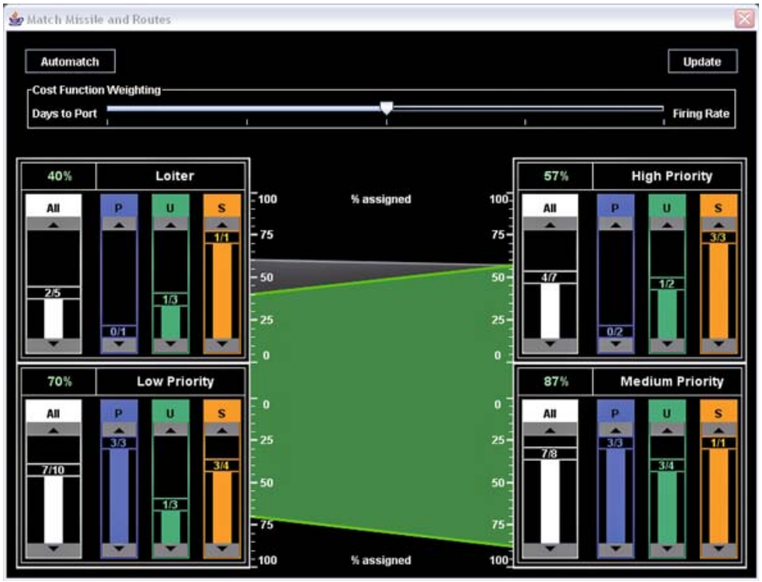


Fig. 26.5 Interface 3

is (2, −1, 2) as neither the moderator nor decider roles changed from interface 2, although the generator's did.

The three interfaces were evaluated with 20 US Navy personnel who would use such a tool in an operational setting. While the full experimental details can be found elsewhere [26.18], in terms of overall performance, operators performed the best with interfaces 1 and 2, which were not statistically different from each other ($p = 0.119$). Interface 3, the one with the predominantly automation-led collaboration, produced statistically worse performance compared with both interfaces 1 and 2 ($p = 0.011$ and 0.031 respectively). Table 26.4 summarizes the HACT categorization for the three interfaces, along with their relative performance rankings. The results indicate that, because the moderator and decider roles were held constant, the degraded performance for those operators using interface 3 was a result of the differences in the generator aspect of the decision-making process. Furthermore, the decline in performance occurred when the LOC was weighted towards the automation. When the solution process was either human-led or of equal contribution, operators performed no differently. However, when the solution generation was automation led, operators struggled.

Table 26.4 Interface performance and HACT three-tuples; M – moderator; G – generator; D – decider

	HACT three-tuple (M, G, D)	**Performance**
Interface 1	(2, 1, 2)	Best
Interface 2	(2, 0, 2)	Best
Interface 3	(2, −1, 2)	Worst

While there are many other factors that likely affect these results (trust, visualization design, etc.), the HACT taxonomy is helpful in first deconstructing the automation components of the decision-making process. This allows for more specific analyses across different *collaboration levels* of humans and automation, which has not been articulated in other LOA scales. In addition, as demonstrated in the previous example, when comparing systems, such a categorization will also pinpoint which LOCs are helpful, or at the very least, not detrimental. In addition, while not explicitly illustrated here, the HACT taxonomy can also provides designers with some guidance on system design, i. e., to improve performance for a system; for example, in interface 3, it may be better to increase the moderator LOC instead of lowering the generator LOC.

In summary, application of HACT is meant to elucidate human–computer collaboration in terms of an information processing theoretic framework. By deconstructing either a single or competing decision support systems using the HACT framework, a designer can better understand how humans and computers are collaborating across different dimensions, in order to identify possible problem areas in need of redesign; for example, in the case of the Patriot missile system with a (−2, −2, 1) three-tuple and its demonstrated poor performance, designers could change the decider role to a 2 (only the human makes the final decision, automation cannot veto), as well as move towards a more truly collaborative solution generation LOC. Because missile intercept is a time-pressured task, it is important that the automation moderate the task, but because of the inability of the automation to always correctly make recommendations, more collaboration is needed across the solution-generation role, with no automation authority in the decider role. Used in this manner, HACT aids designers in the understanding of the multiagent roles in human–computer collaboration tasks, as well as identifying areas for possible improvement across these roles.

26.4 Conclusion and Open Challenges

The human–automation collaboration taxonomy (HACT) presented here builds on previous research by expanding the *Parasuraman* [26.1] information processing model, specifically the decision-making component. Instead of defining a simple level of automation for decision making, we deconstruct the process to include three distinct roles, that of the moderator (the agent that ensures the decision-making process moves forward), the generator (the agent that is primarily responsible for generating a solution or set of possible solutions), and the decider (the agent that decides the final solution along with veto authority). These three distinct (but not necessarily mutually exclusive) roles can each be scaled across five levels indicating degrees of collaboration, with the center value of 0 in each scale representing balanced collaboration. These levels of collaboration (LOCs) form a three-tuple that can be analyzed to evaluate system collaboration, and possibly identify areas for design intervention.

As with all such levels, scales, taxonomies, etc., there are limitations. First, HACT as outlined here does not address all aspects of collaboration that could be considered when evaluating the collaborative nature of a system, such as the type and possible latencies in communication, whether or not the LOCs should be dynamic, the transparency of the automation, the type of information used (i. e., low-level detail as opposed to higher, more abstract concepts), and finally how adaptable the system is across all of these attributes. While this has been discussed in earlier work [26.7], more work is needed to incorporate this into a comprehensive yet useful application.

In addition, HACT is descriptive versus prescriptive, which means that it can describe a system and identify post hoc where designs may be problematic, but cannot indicative how the system should be designed to achieve some predicted outcome. To this end, more research is needed in the application of HACT and the interrelation of the entries within each three-tuple, as well as more general relationships across three-tuples. Regarding the within three-tuples issue, more research is needed to determine the impact and relative importance of each of the three roles; for example, if the moderator is at a high LOC but the generator is at a low LOC, are there generalizable principles that can be seen across different decision support systems? In terms of the between three-tuple issue, more research is needed to determine under what conditions certain three-tuples produce consistently poor (or superior) performance, and whether these are generalizable under particular contexts; for example, in high-risk time-critical supervisory control domains such as nuclear power plant operations, a three-tuple of (−2, −2, −2) may be necessary. However, even in this case, given flawed automated algorithms such as those seen in the Patriot missile, the question could be raised of whether it is ever feasible to design a safe (−2, −2, −2) system.

Despite these limitations, HACT provides more detailed information about the collaborative nature of systems than did previous level-of-automation scales, and given the increasing presence of *intelligent* automation both in complex supervisory control systems and everyday life, such as global positioning system (GPS) navigation, this sort of taxonomy can provide for more in-depth analysis and a common point of comparison across competing systems. Other future areas of research that could prove useful would be the determination of how levels of collaboration apply in the other data acquisition and action implementation information processing stages, and what the impact on human performance would be if different collaboration levels were mixed across the stages. Lastly, one area often overlooked that deserves much more attention is the ethical and social impact of human–computer collaboration. Higher levels of automation authority can reduce an operator's awareness of critical events [26.19] as well as reduce their sense of accountability [26.20]. Systems that promote collaboration with an automated agent could possibly alleviate the offloading of attention and accountability to the automation, or collaboration may further distance operators from their tasks and actions and promote these biases. There has been very little research in this area, and given the vital nature of many time-critical systems that have some degree of human–computer collaboration (e.g., air-traffic control and military command and control), the importance of the social impact of such systems should not be overlooked.

References

26.1 R. Parasuraman, T.B. Sheridan, C.D. Wickens: A model for types and levels of human interaction with automation, IEEE Trans. Syst. Man Cybern. – Part A: System and Humans **30**(3), 286–297 (2000)

26.2 P. Dillenbourg, M. Baker, A. Blaye, C. O'Malley: The evolution of research on collaborative learning. In: *Learning in Humans and Machines. Towards an Interdisciplinary Learning Science*, ed. by P. Reimann, H. Spada (Pergamon, London 1995) pp. 189–211

26.3 J. Roschelle, S. Teasley: The construction of shared knowledge in collaborative problem solving. In: *Computer Supported Collaborative Learning*, ed. by C. O'Malley (Springer, Berlin 1995) pp. 69–97

26.4 P.J. Smith, E. McCoy, C. Layton: Brittleness in the design of cooperative problem-solving systems: the effects on user performance, IEEE Trans. Syst. Man Cybern. **27**(3), 360–370 (1997)

26.5 H.A. Simon, G.B. Dantzig, R. Hogarth, C.R. Plott, H. Raiffa, T.C. Schelling, R. Thaler, K.A. Shepsle, A. Tversky, S. Winter: Decision making and problem solving, Paper presented at the Research Briefings 1986: Report of the Research Briefing Panel on Decision Making and Problem Solving, Washington D.C. (1986)

26.6 P.M. Fitts (ed.): *Human Engineering for an Effective Air Navigation and Traffic Control system* (National Research Council, Washington D.C. 1951)

26.7 S. Bruni, J.J. Marquez, A. Brzezinski, C. Nehme, Y. Boussemart: Introducing a human–automation collaboration taxonomy (HACT) in command and control decision-support systems, Paper presented at the 12th Int. Command Control Res. Technol. Symp., Newport (2007)

26.8 M.P. Linegang, H.A. Stoner, M.J. Patterson, B.D. Seppelt, J.D. Hoffman, Z.B. Crittendon, J.D. Lee: Human–automation collaboration in dynamic mission planning: a challenge requiring an ecological approach, Paper presented at the Human Factors and Ergonomics Society 50th Ann. Meet., San Francisco (2006)

26.9 Y. Qinghai, Y. Juanqi, G. Feng: Human–computer collaboration control in the optimal decision of FMS scheduling, Paper presented at the IEEE Int. Conf. Ind. Technol. (ICIT '96), Shanghai (1996)

26.10 R.E. Valdés-Pérez: Principles of human–computer collaboration for knowledge discovery in science, Artif. Intell. **107**(2), 335–346 (1999)

26.11 B.G. Silverman: Human–computer collaboration, Hum.–Comput. Interact. **7**(2), 165–196 (1992)

26.12 L.G. Terveen: An overview of human–computer collaboration, Knowl.-Based Syst. **8**(2–3), 67–81 (1995)

26.13 G. Johannsen: *Mensch-Maschine-Systeme (Human–Machine Systems)* (Springer, Berlin 1993), in German

26.14 J. Rasmussen: Skills, rules, and knowledge; signals, signs, and symbols, and other distractions in human performance models, IEEE Trans. Syst. Man Cybern. **13**(3), 257–266 (1983)

26.15 T.B. Sheridan, W. Verplank: *Human and Computer Control of Undersea Teleoperators* (MIT, Cambridge 1978)

26.16 M.R. Endsley, D.B. Kaber: Level of automation effects on performance, situation awarness and workload in a dynamic control task, Ergonomics **42**(3), 462–492 (1999)

26.17 V. Riley: A general model of mixed-initiative human–machine systems, Paper presented at the Human Factors Society 33rd Ann. Meet., Denver (1989)

26.18 S. Bruni, M.L. Cummings: Tracking resource allocation cognitive strategies for strike planning, Paper presented at the COGIS 2006, Paris (2006)

26.19 K.L. Mosier, L.J. Skitka: Human decision makers and automated decision aids: made for each other? In: *Automation and Human Performance: Theory and Applications*, ed. by R. Parasuraman, M. Mouloua (Lawrence Erlbaum, Mahwah 1996) pp. 201–220

26.20 M.L. Cummings: Automation and accountability in decision support system interface design, J. Technol. Stud. **32**(1), 23–31 (2006)