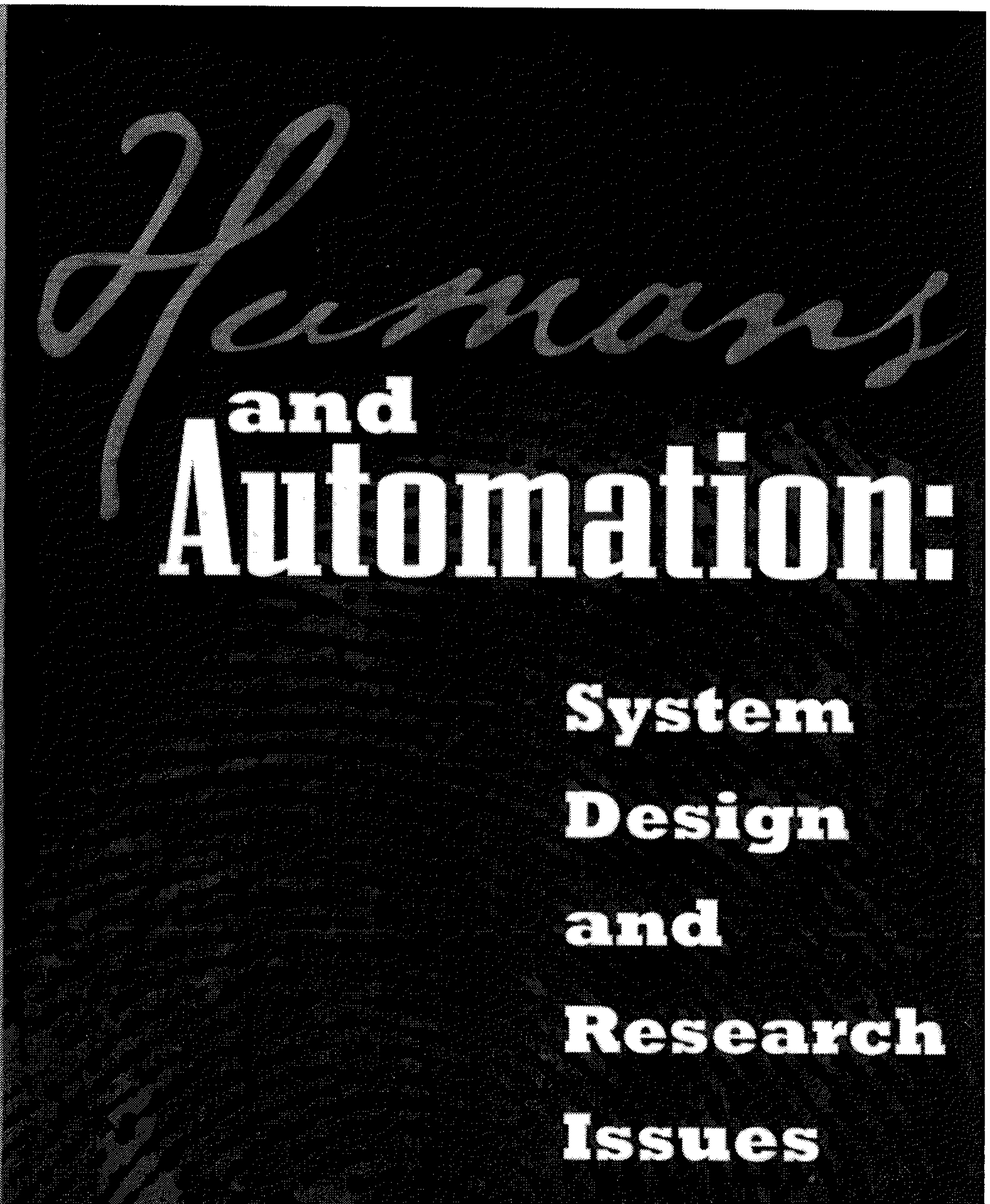
Chapter 3



Humans and Automation:

System Design and Research Issues

Thomas B. Sheridan

Wiley Series in System Engineering and Management
HFES Issues in Human Factors and Ergonomics Series

Chapter 3

THE ANALYSIS AND DESIGN PROCESS

The Role of the Human Factors Professional

The purpose of design, from the point of view of the human factors professional, is to achieve a proper relationship between the behavior of the hardware and software technology and the behavior of the human user. Many engineers have been conditioned to think that design is always design of a physical thing, so design of a relationship to a human user is a difficult concept.

Human factors professionals are trained to start from the viewpoint of the human user (and from the fact that designer "common sense" may be flawed). Their training is varied, as some start from an engineering base, some from an experimental psychology or physiology base, and some from a base of medicine, dentistry, or another applied human science. The common viewpoint is an orientation of experimental science, an appreciation for the variability among people and the variability of behavior within the same person. They also understand the use of statistics to make rational inferences from observations, and they have the ability to apply mathematical modeling where appropriate. There are boards of certification for human factors professionals, but currently there is no consensus on whether certification is an important pedigree.

The human factors professional will usually insist on performance tests using either the real environment (field tests) or some type of simulation (if the system under consideration is not yet in the real world). He or she will typically ask many questions that may seem antagonistic to the design engineer, who often assumes the protagonist role for any new design; nevertheless, taking an initially neutral stance toward the proposed benefits of any new design is a sacred obligation of the human factors professional. For these reasons it is important to bring the human factors viewpoint into consideration early in the design process, not later, after the design has been frozen and there is little real opportunity to correct any shortcomings that are found.

Human factors professionals should not be segregated from the design engineers. Human factors types may feel more comfortable maintaining their own identity in their separate ivory towers, surrounded by their own laboratories and their own jargon. However, they need to expose themselves to and learn the jargon and the ways of thinking of the design engineers, bringing the user representatives along into that same environment, as appropriate. Only then will they win the respect of the engineers and see their advice implemented. Simply publishing research results in learned journals and posting human factors guidelines (much as Martin Luther posted his ultimatums on the church door) often leads to being ignored by the engineers.

Accident data banks are available and can be very helpful. Anonymous self-reports of aircraft near-misses by pilots have resulted in a widely published NASA/FAA database. The U.S. Department of Transportation has, over many years, compiled an extensive database of fatal auto accidents. Confidential accident and incident data collected within industry, NASA, FAA, the Department of Transportation, Department of Defense, Nuclear Regulatory Commission, and elsewhere can easily be sanitized and made available for general use. Although every accident or near-accident is unique in some respect, with enough data, some powerful and useful inferences can be drawn.

Human-Machine Task Analysis: Goals and Constraints

Task analysis is commonly regarded as the first step in the human-system design process. Task analysis requires the breakdown of overall tasks into their elements and the specification of how these elements relate to one another in space, time, and functional relation. Who (including which person, if there are several, or which machine) does what, when, and how? By *task* one can mean the complete performance of a given procedure, the totality of activities to design and/or build a given thing, to monitor and/or control a given system, or to diagnose or solve a given problem. Alternatively, *task* can mean a small subelement, such as a particular movement or measurement. Sometimes *mission* is used to connote the full task or end state to be achieved, and *subtask* is used to refer to some component. Terminology for such breakdown of tasks is of no particular concern here except to note that some hierarchical breakdown is usually helpful or even necessary.

When a system is being automated, or existing automation is being improved, it is particularly important to rethink the task analysis. This is because automation can change the nature of demands and responsibilities, often in ways that were unintended or unanticipated (Boy, 1998; Parasuraman, Sheridan, & Wickens, 2000; Parasuraman & Riley, 1997; Wiener & Curry, 1980; Woods, 1996).

The starting point is usually a set of implicit task constraints, and task analysis is really a matter of articulating these constraints and making them visible, being very clear about what are independent and what are dependent variables.

Although many techniques are used for task analysis (Kirwan & Ainsworth, 1992), it is still very much an art. Task analyses end up being verbal statements of missions with qualifications, mathematical equations, block diagrams defining elements with arrows indicating what influences what, lists of variables with ranges and/or statistical properties, timelines, flow charts, and so forth.

When performing a task analysis, it is all too easy to assume more constraints than are really there or are necessary. For example, there is a tendency to fall into the trap of writing down the series of steps a human takes to look at particular already existing displays and operate particular already existing controls, the very displays and controls that constitute the present interface one seeks to improve or to automate.

The correct way to analyze the task is to specify the information required, the decisions to be made, the control actions to be taken (at the level of the controlled process), and the criterion for satisfactory completion of that step (Figure 3.1). This should be independent of the particular human or machine means to achieve those steps — that is, the particular displays or controls that already are, or even those that might be, used for task implementation.

Following the ecological perspective (Vicente, 1999; see the section on ecological displays in Chapter 5 of this book) and the idea of *joint cognition* (Dowell & Long, 1998; Hollnagel & Woods, 1983; Rasmussen, Pejtersen, & Goodstein, 1994; Woods & Roth, 1995), there is a call for postponing the traditional task analysis until after the work domain constraints have been thoroughly spelled out. Studying and specifying work constraints, it seems to me, is an obvious part of defining any problem, and if any task analysis avoids this, it is to be faulted.

Task step	Operator or machine identification	Information required	Decision(s) to be made	Control action required	Criterion of satisfactory completion

Figure 3.1 An example format for task analysis.

Human interaction with the environment is a closed-loop process: The human observes the environment and in turn affects the environment. Causation goes both ways, as depicted in Figure 3.2. Therefore it is important to think of the constraints that operate in both directions: all the factors that constrain the human from observing the environment fully and perfectly, and all the factors that constrain the human from action upon the environment exactly as he or she might wish. In Figure 3.2 these constraints are subsumed under the rubric *filters*. It is important to keep track of which factors act as filters or constraints in the human observation (affective) direction and which act in the human motor action (effective) direction.

It is not easy to do the task analysis without implying some *function allocation* — what functions should be allocated to humans and what functions to machines — which is really part of system synthesis. Nevertheless, the effort must be made to determine what is necessarily a given and what may be left as degrees of freedom for synthesis.

Human-Machine Function Allocation

Function allocation requires the designer to assign various functions (tasks, jobs) that need to be done (as specified in the task analysis) to resources — that is, to particular humans, instruments, or machine agents capable of doing those tasks. Alternatively, function allocation can be thought of as the allocation of such resources to tasks, which is equivalent.

Many tasks must be done by a human because it is not known how to automate them. Other tasks are done by humans because humans like to do them. Still others are done by humans because the tasks are trivial or are embedded in either of the other two types of task, and therefore it is inconvenient to devise some automation to do them. To the extent that a task now done by a human is boring, fatiguing, or hazardous, it should be automated if feasible and convenient to do so. Otherwise, when a task can

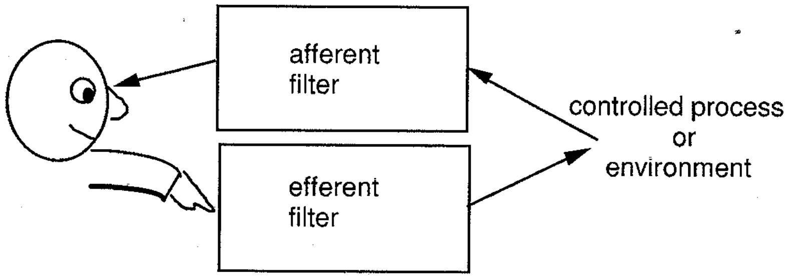


Figure 3.2 Afferent and efferent filters between human and environment.

be done by either human or machine, it is worth considering whether human and machine might work together and complement each other.

Currently, although many accepted techniques exist for doing task analysis, there is no commonly accepted means to perform function allocation (and especially to "optimize" this allocation, a bit of semantic nonsense that one sometimes sees in print). The reasons are several. First, although tasks/functions (the "whats" and the "hows") may indeed be broken into pieces, those pieces are seldom independent of one another, and the task components may interact in different ways depending on the resources chosen for doing them. Second, the human and computer can interact in an infinite number of ways, resulting in an infinite spectrum of allocation possibilities from which to choose. Third, criteria for judging the suitability of various human-machine mixes are usually difficult to quantify and often implicit. Although I suggest some things to consider when allocating tasks to people and machines (especially machines with so-called intelligent capabilities), I make no attempt to offer a general procedure for synthesizing the task allocation.

The Fitts MABA-MABA List

Fitts (1951) proposed a list of what "men are better at" and what "machines are better at" (MABA-MABA; see Table 3.1). This is sometimes called the Fitts MABA-MABA list or simply the Fitts list. It is often considered to be the first well-known basis for task allocation. An expanded version appears in NUREG-0700 (U.S. Nuclear Regulatory Commission, 1981).

Jordan (1963) was concerned that the Fitts list might be used by people who assume that the goal is to compare people with machines and then to decide which is best for each function or task element. He quoted Craik (1947a, 1947b), who had pointed out that to the extent that the human is understood as a machine, it can be known how to replace a human with a machine. Early studies by Birmingham and Taylor (1954) revealed that in simple manual control loops, performance can be improved by quickening, wherein visual feedback signals are biased by derivatives of those signals, thereby adding artificial anticipation (what the control engineer would call *proportional-plus-derivative control*) and saving the human operator the trouble of performing this computation cognitively. Birmingham and Taylor concluded that "man is best when doing least" in this case.

TABLE 3.1
The Fitts MABA-MABA List

Men Are Better At	Machines Are Better At
Detecting small amounts of visual, auditory, or chemical energy	Responding quickly to control signals
Perceiving patterns of light or sound	Applying great force smoothly and precisely
Improvising and using flexible procedures	Storing information briefly, erasing it completely
Storing information for long periods of time and recalling appropriate parts	Reasoning deductively
Reasoning inductively	
Exercising judgment	

Jordan suggested that this is a perfect example of Craik's tenet. He also quoted Einstein and Infeld (1942), who discussed the development and then the demise of the concept of ether in physics and how, when empirical facts do not agree with accepted concepts, it is time to throw out the accepted concepts (but retain the empirical facts). Jordan's point was that the idea of comparing the human with the machine should be thrown out but that the facts about what people do best and what machines do best should be retained. Jordan's idea of what should be espoused, and the main point of retaining the Fitts list, is that people and machines are complementary.

Also in keeping with Craik's (1947a, 1947b) tenet, Jones and Jasek (1997), Woods (1996), and Hoc (2000) emphasized that the goal is human-machine cooperation — not the allocation of separate functions to separate entities — and this includes cooperation in planning and action. However, exactly what it means to cooperate is still to be worked out in terms useful for human factors engineering.

It might seem that a straightforward procedure for doing task allocation would be to write down all the salient mixes of allocation and then write down all the applicable criteria. Following this, one might rank order all combinations of allocation mix and criteria, thus determining a rank-order score. Alternatively, one could weight the relative importance of each criterion, rate each mix by each criterion, multiply by the weight, and add up the scores for each allocation mix. However, there can be difficulties with any such direct method — hidden assumptions, unanticipated criteria situations, nonindependence of tasks, nonindependence of criteria, nonlinearities that invalidate simple multiplication of weight by rating and addition of products — in particular, the fact that a very large number of possible interactions between human and computer compete for consideration; it is not a simple matter of "human versus computer," a point that is emphasized repeatedly in this book.

Price (1985) asserted that in order to make use of the Fitts MABA-MABA list, one needs data that are context dependent, but these data are mostly unavailable. The difficulty in acquiring these data is exacerbated by the fact that the status of the machine is not static; the capabilities of machines to perform intelligent acts such as automation and decision support are ever improving. However, Price claimed, automation can "starve cognition" if the human is not kept in sufficient communication with what the automation is doing or intending. He seems to have agreed with Jordan (1963) when he pointed out that human performance and machine performance are not a zero-sum game, implying that the combination can be much better than either by itself. Kantowitz and Sorkin (1987), Price (1990), and Volume 52 of the *International Journal of Human-Computer Studies* (2000) provide reviews of the literature in task allocation.

One can make a distinction between tasks as goals to be achieved and functions as means to achieve the goals, what the Fitts list specifies as best for each of human and machine. If two functions within a task are closely related and interdependent, it may be inadvisable to allocate one to the machine and one to the human. For example, for air traffic control, Bisseret (1970) recommended against systematically assigning the conflict detection function to the machine and the resolution function to the human, suggesting rather that the pair of functions be assigned to either human or machine depending on the type of conflict. In other words, assignment should be by task rather than by function (Vanderhaegen, Crevits, Debernard, & Millot, 1994).

According to Rasmussen (2000), human-automation design should not be based on normative work procedures but, rather, should allow a "resource envelope" within which the operator can work freely and still maintain the support of the automation.

The Fitts list generalizations remain a useful starting point for allocation of functions between human and computer. No other allocation model has replaced it in terms of simplicity and understandability. However, it is a qualitative statement subject to interpretation. As computers become "smarter," they creep closer to bettering human capabilities. One can say for sure, based on lots of evidence, that human memory prefers large chunks with interconnected or associated elements, prefers relatively complete pictures and patterns, and tends to be less good at details. This empirical fact strongly suggests that insofar as it is feasible and practical, the human should be left to deal with the "big picture" while the computer copes with the details.

Whether function allocation will ever become more a science than an art is debatable. In fact, the issue has sparked much recent debate. Fuld (2000) advocated a bottom-up rather than a top-down approach, believing that function allocation is a useful theory but not a practical method. In Sheridan (2000), I asserted that the human-automation mix will necessarily and unstoppably evolve because of forces of technology and culture, but that does not excuse the designer from doing his or her best, especially when human values are at stake.

The public (as well as, unfortunately, too many political and industry decision makers) has been slow to realize that function/task allocation does not usually mean allocation of a whole task to either human or machine, exclusive of the other. For example, in the space program it has been common to consider that a task must be done by either an astronaut or a robot; that if a spacecraft is staffed, then astronauts must do almost everything; and that if a spacecraft is unstaffed, every task must be automated. In fact, on occupied spacecraft many functions are automatic, and on unoccupied spacecraft many functions are performed by human remote control from the ground. In the next section I show that different stages of a task can be separately automated to greater and lesser degrees.

Allocation at Four Stages and in Differing Degrees

When considering how far to go with automation and decision aiding, the system designer should ask what parts of the system cannot be automated because it is not known how to automate them, what parts humans like to do, what parts are trivial for humans and inconvenient to automate, and what parts are boring, fatiguing, or hazardous and thus should be automated if possible. Also, what degree of unfamiliarity can humans cope with, and what degree of unfamiliarity can be accommodated by automation using the best available predictive models? Surely automation need not apply uniformly to entire tasks; rather, it can be of more benefit to some parts than to others.

A taxonomy that applies to most complex human-machine systems is the sequence of operations shown in Figure 3.3: acquire information,

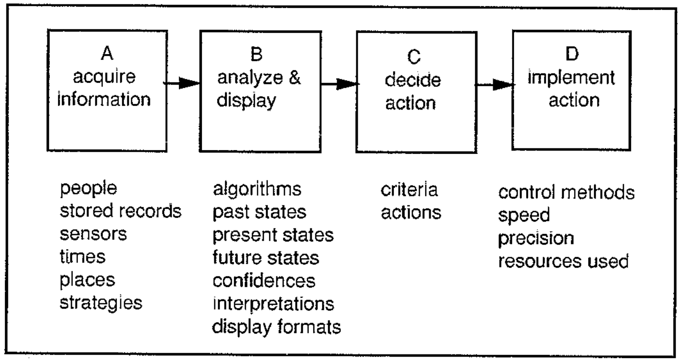


Figure 3.3 Four stages of a complex human-machine task.

analyze and display, decide action, and implement action. Below each box in this figure are noted some typical variables relevant to that stage.

Table 3.2 presents an eight-level scale of degrees of automation, a simplification of an earlier scale (Sheridan & Verplank, 1978). To make it simple, the scale folds together several relevant dimensions, such as

- the degree of specificity required of the human for inputting requests to the machine,
- the degree of specificity with which the machine communicates decision alternatives or action recommendations to the human;
- the degree to which the human has responsibility for initiating implementation of action, and
- the timing and detail of feedback to the human after machine action is taken.

It is evident that these many dimensions of automation can be ordered in various ways on multidimensional as well as one-dimensional scales. The main point is to have some way of considering many options that are more or less ordered and then to decide the degree of automation at each stage. There is no reason to force each stage to the same level of automation.

Figure 3.4 illustrates how different kinds of tasks (represented by the three types of lines) call for different levels of automation at different stages. The numbers correspond to the eight-point scale, Level 1 representing the *least automation* and Level 8 the *most automation.* The circles might represent, for example, the task of holding an election of officers for an organization (Sheridan, 1998). At the *acquire* stage, data are collected manually from eligible individuals, although e-mail (Level 2 automation) might have been used to make suggestions to solicitors on how to operate. The results are then *analyzed* by computer, and the winners are *decided* automatically. The transfer of power is then *implemented* with only a modest amount of computer-stored procedural advice.

TABLE 3.2
A Scale of Degrees of Automation

1. The computer offers no assistance; the human must do it all.
2. The computer suggests alternative ways to do the task.
3. The computer selects one way to do the task and
4. executes that suggestion if the human approves, or
5. allows the human a restricted time to veto before automatic execution, or
6. executes automatically, then necessarily informs the human, or
7. executes automatically, then informs the human only if asked.
8. The computer selects the method, executes the task, and ignores the human.

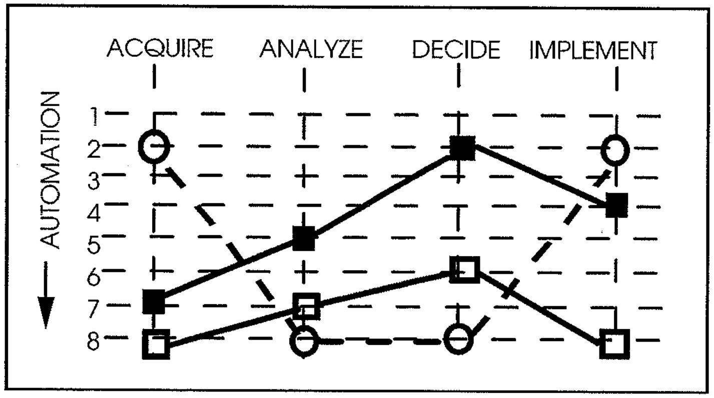


Figure 3.4 Examples of different levels of automation at different task stages.

The black squares might represent advice recently given by a U.S. National Research Council (Wickens, Mavor, et al., 1998) committee advising on the proper amount of automation for new civilian ATC systems. After much debate, the committee decided that acquisition and analysis could and should be highly automated — in fact, they already are (radar, weather, schedule information, etc.) — but that decision making, except for certain decision aids now under development, should be done by human air traffic controllers. Implementation is in the hands of the pilots, which in turn is largely turned over to autopilots and the other parts of the Flight Management System.

The open squares might represent a typical manufacturing robotics task. A computer vision system acquires all the data. This is analyzed in a computer where the analysis results are available to a human supervisor should he or she need to check the data. The analysis is passed on to a computer decision algorithm, and the decision results are displayed for the operator. The decision is implemented by a robot in fully automatic fashion.

As part of function allocation, Wei, Macwan, and Wierenga (1998) suggested a model for the appropriate degree of automation of different tasks, based on a task's effect on system performance and its demand on the operator relative to other tasks. Experimentally they showed that as automation was increased, performance did not improve beyond a certain level.

Human-Computer Trading and Sharing

Human and computer can either pass control alternately back and forth (called *trading*) or can function simultaneously (called *sharing* the task).

Both trading and sharing may occur as parts of the same overall task. Figure 3.5 conveys the idea: The connections in the top part of the figure alternate between human and computer, whereas in the bottom part they remain fixed in parallel operation.

Trading is necessary when the human must do some of the task and then turn it over to the computer to do other parts. For example, developing software involves both programming and testing alternately. Trading is often recommended when the human or machine performs a task and the other does a check. It can also be useful if the one that normally does a task is busy and the other can step in on a temporary basis to settle urgent matters or "buy time."

Sharing is useful when human and machine can do different portions of the same task — for instance, if the human controls in some degrees of freedom while the machine simultaneously controls in others. This may be necessary in some real-time control or monitoring tasks, or it may be a matter of making the best use of the human and machine resources.

Adaptive Automation

A number of researchers have pointed out the need for *adaptive automation*, meaning that the allocation of functions between human and automation is not fixed but varies with time depending on the human's momentary workload as well as the current context (Hancock & Scallen, 1996; Kaber & Riley, 1999; Moray, Inagaki, & Itoh, 2000; Parasuraman, Mouloua, & Molloy, 1996; Rouse, 1988). A change in function allocation could also be at the request of the human (asking for help) or at the insistence of the automation (telling the human the job is not getting done, or simply taking over when the human is napping or otherwise preoccupied).

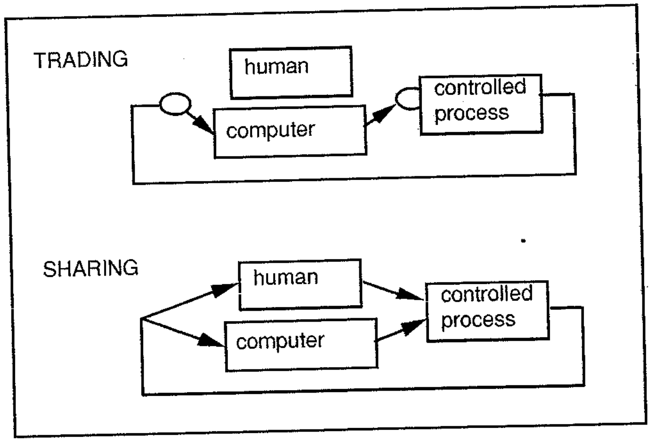


Figure 3.5 Trading and sharing.

Large System Organization

In the preceding sections I have discussed task analysis and function allocation at the level of the workplace for a single human operator or a small team. When many machines and people make up an operating system, the task analysis and function allocation must be viewed at a higher level of system organization (as well as at the lower level) with respect to both spatial allocation (building architecture, traffic system structure, etc.) and temporal dynamics.

A good example is the organization of a modern factory. Modern manufacturing is driven by cost, and major determinants of cost are wastage of human operator time and wastage of parts inventory. For this reason, the organization in space (factory layout) and the organization in time (work scheduling) are critical. One normally thinks of a factory as a system that processes raw materials to make parts, which are then assembled to make products, which are then shipped to the customer (which might itself be a conventional wholesaler, an Internet wholesaler, a retail store, or an individual). This material flow is shown from left to right in Figure 3.6. What has often been neglected in the past is the information (requirements) flow (right to left in the figure). Three feedback loops are operative: between raw materials and material processing operations (machining, forming, welding, etc.); between material processing and parts assembly, and between parts assembly and customer demand. Each information requirement provides a set point to the control loop to its left.

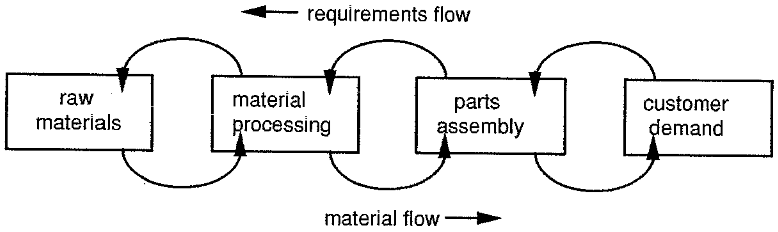


Figure 3.6 Basis for organizing a factory in space and time.

Optimization

The system designer would like to have the goal(s) or objectives of the system specified quantitatively, stated as relative goodness of performance (a scalar) as a function of system inputs and outputs and resources used. This is called an *objective function* or *utility function* (Keeney & Raiffa, 1976). (Objective functions and the trade-off problem are discussed formally in Appendix A.) This is also true of given physical and economic constraints, such as size, force, time, and money.

The controlled process – that is, the relations between independent and dependent variables of the physical thing being controlled (the aircraft, automobile, nuclear reactor, etc.) – will likely already have been modeled, or at least approximated, mathematically, for formal closed-loop control requires at least an estimated process model. The automatic controller can also be modeled if it is simple enough; for instance, if it is a sum of differently weighted proportional, time-derivative, and time-integral functions of the control error (system state deviation from what is wanted), such as make up linear control theory (see Appendix A).

In the engineer's ideal world, one could write explicit equations for (a) the controlled process, (b) the constraints (physical, economic, and human), and (c) the objective function. Then, by simultaneous solution of these equations, one might maximize "goodness," and thus an optimal control strategy would be determined. This, in effect, is what is done in optimal control theory, linear and dynamic programming, and myriad other mathematical techniques for optimization. The reason these techniques are seldom applied to real-world systems is that the assumptions cannot be fulfilled. One simply does not have the necessary information in explicit mathematical form.

Given that real-world optimization of purely physical automation systems is sometimes tractable (because a mathematical objective function is available to specify what is better and what is worse), what about human-automation systems? Can they be optimized? Here it has to be asserted that optimization in the formal sense is simply out of the question, because the cognitive contribution of the human operator can hardly be put in mathematical form. Thus claims or stated intentions of optimization of a human-machine system are meaningless, at least in the formal sense that optimization is a unique state.

What is much more likely is gradual improvement of systems relative to (sometimes changing) criteria — attributable to iterative redesign and operator/user experience over a long period of evolution (Sheridan, 2000).

Handbooks, Guidelines, and Standards

Over the half-century that human factors engineering has existed, various handbooks on that topic have been published, probably the best known of which are Van Cott and Kinkade (1972), Boff and Lincoln (1988), Salvendy (1997), and Helander, Landauer, and Prabhu (1997). Many handbooks are to be found in the automation technology field, such as Shell and Hall (2000). Many of the same topics are covered in handbooks of the traditional engineering disciplines; for example, in mechanical engineering, see Kreith (1997).

Particular industries, or government agencies that support or regulate industries, have often published design guidelines (e.g., see NUREG-0700, U.S. Nuclear Regulatory Commission, 1981, for nuclear plant control room design). In the case of either handbooks or guidelines, the user must be careful to question whether any particular bit of advice, whether implied or stated explicitly, applies in the particular context of the current problem he or she is seeking to solve. Researchers like to generalize their results, and readers often are too ready to accept those generalizations without qualifying them in terms of the particular experiment or design constraints in which they were presented. Sometimes authors preparing design guidelines for one context have simply incorporated into their works the already published guidelines for another design context.

If guidelines are to be adapted from one application context into another, it is mandatory that they be reviewed with the new context in mind. In one case, for a nuclear plant control room alarm, I encountered a government-issued guideline for sound intensity level (in decibels) that seemed rather too loud. Upon investigation I discovered that the specification was lifted verbatim from a recommendation for military aircraft in which the pilots wore earphones and the sound source was on an external panel, so the alarm had to be loud enough to penetrate the sound-barrier ear cups on the earphones. Nuclear plant operators, however, don't wear earphones. The alarm level recommended by the guidelines would almost have deafened these operators.

Standards are a different story. Standards emanate from well-respected organizations such as the International Organization for Standardization (ISO), the American National Standards Institute (ANSI), and the Institute of Electrical and Electronics Engineers (IEEE). Standards committees are formed around particular problem areas by enlisting volunteers from various companies or other organizations representative of that industrial sector. These committees gather in multiple meetings to discuss in technical detail what consensus might be reached on design particulars. When a committee reaches consensus, it publishes a draft standard that is circulated throughout the industry for comment. Based on comments received, the standard is revised, and after several iterations a final standard is published. At that point the standard has real power, because a manufacturer of a product that deviates from the standard may have difficulty getting its products marketed, getting suppliers to carry spare parts, and so forth. Furthermore, tort lawyers can more easily claim so-called design defects if a product's design is shown to have ignored a published industry standard.

Standards making is not always a neat and clean exercise. Stewart (2000) reviewed the status of ISO 9241 (ergonomic requirements for office work with visual display terminals). According to Stewart, users of this standard have had unrealistic expectations that it would solve all their problems. Furthermore, the technology has changed so fast that the committee deliberations have not been able to keep up, and the recommendations that were proffered were often based on the national or industrial interests of the participants rather than the interests of potential users. Nevertheless, in spite of the politics of standards promulgation, product and system designers can obtain much that is useful from standards, along with other information sources such as handbooks, the general scientific literature, and qualified human factors experts.