SEMI-STRUCTURED DECISION PROCESSES: A CONCEPTUAL FRAMEWORK FOR UNDERSTANDING HUMAN-AUTOMATION DECISION SYSTEMS

William N. Kaliardos and R. John Hansman

Report No. ICAT-99-7
September 1999

MIT International Center for Air Transportation
Department of Aeronautics & Astronautics
Massachusetts Institute of Technology
Cambridge, MA 02139 USA

SEMI-STRUCTURED DECISION PROCESSES: A CONCEPTUAL FRAMEWORK FOR UNDERSTANDING HUMAN-AUTOMATION DECISION SYSTEMS

WILLIAM N. KALIARDOS AND R. JOHN HANSMAN, JR.

Abstract

The purpose of this work is to improve understanding of existing and proposed decision systems, ideally to improve the design of future systems. A "decision system" is defined as a collection of information-processing components -- often involving humans and automation (e.g., computers) -- that interact towards a common set of objectives. Since a key issue in the design of decision systems is the division of work between humans and machines (a task known as "function allocation"), this report is primarily intended to help designers incorporate automation more appropriately within these systems.

This report does not provide a design methodology, but introduces a way to qualitatively analyze potential designs early in the system design process. A novel analytical framework is presented, based on the concept of "semi-Structured" decision processes. It is believed that many decisions involve both well-defined "Structured" parts (e.g., formal procedures, traditional algorithms) and ill-defined "Unstructured" parts (e.g., intuition, judgement, neural networks) that interact in a known manner. While Structured processes are often desired because they fully prescribe how a future decision (during "operation") will be made, they are limited by what is explicitly understood prior to operation. A system designer who incorporates Unstructured processes into a decision system understands which parts are not understood sufficiently, and relinquishes control by deferring decision-making from design to operation. Among other things, this design choice tends to add flexibility and robustness. The value of the semi-Structured framework is that it forces people to consider system design concepts as operational decision processes in which both well-defined and ill-defined components are made explicit. This may provide more insight into decision systems, and improve understanding of the implications of design choices.

The first part of this report defines the semi-Structured process and introduces a diagrammatic notation for decision process models. In the second part, the semi-Structured framework is used to understand and explain highly evolved decision system designs (these are assumed to be representative of "good" designs) whose components include feedback controllers, alerts, decision aids, and displays. Lastly, the semi-Structured framework is applied to a decision system design for a mobile robot.

This document is based on the thesis of William Kaliardos submitted in partial fulfillment of the degree of Doctor of Philosophy in Aeronautics and Astronautics at the Massachusetts Institute of Technology, September 1999.

Acknowledgements

This work was supported by the Charles Stark Draper Laboratory, Inc., under IR&D effort 1015. Publication of this report does not constitute approval by Draper or the sponsoring agency of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

Table of Contents

CHAPTER FOUR THE DESIGN OF A DECISION SYSTEM FOR A MOBILE ROBOT

CHAPTER FIVE SUMMARY AND CONCLUSIONS

CHAPTER ONE

INTRODUCTION

1.1 MOTIVATION

An important issue in the design of decision systems is the manner in which work is divided between people and machines. This design task is known as function allocation. While it is often recognized that there is potential system value to some mix of human and automation, there is no single accepted strategy that determines this mix [139].

Many researchers agree that at the *formal* extreme of allocation methods, a purely algorithmic design process is not likely to be effective [116], [129]. Yet on the *informal* extreme, function allocation can be improved beyond the intuitive, *ad hoc* approaches that are often taken in the absence of formal methods.

It is desired to improve function allocation process by reducing the *ad hoc* component of design, and without resorting to a prescriptive design methodology.

1.2 SCOPE

The concepts in this thesis are not directed towards a single specific system. Because of its generality, it may be helpful to define some common terms and typical applications.

In this thesis, "automation" or "machine" is implied to be for *information* processing within a decision system—typically with a digital computer—and not for performing physical work. A few example uses of automation within a human-automation system are:

- Displays

- Alerts
- Decision aids
- Failure diagnosis
- Control of physical systems

A "decision system" means a collection of distinct parts—generally humans and/or machines—that interact to make decisions in the pursuit of a common objective, as defined by the system function. In this thesis, a decision system is not limited to any particular application, although specific representative systems are analyzed. *Typical* applications of human-automation decision systems are vehicles (e.g., aircraft, spacecraft, automobiles), manufacturing systems, power generation plants (e.g., nuclear), military command and control, urban planning, corporate management, and engineering design.

1.3 BACKGROUND

1.3.1 Prescriptive Methods for Function Allocation

The allocation of functions between humans and machines by formula, algorithms, or other prescriptive methods has shown little success in practice. Various prescriptive methods for function allocation between humans and machines have been proposed [69]. After *mandatory* allocation (e.g., based on physical limitations or policy), Bailey [10] categorizes these into three groups:

- Comparison allocation
- Leftover allocation
- Economic allocation

In *comparison allocation*, humans and machines are directly compared based on a particular sub-function. The designer judges whether a human or machine is superior at a given task, assisted by pre-defined lists of similar functions in which this evaluation has already been made. The "Fitts list" [44] seems to be the most cited, but many similar "MABA" ("men are better at / machines are better at") lists have since been made [116].

In *leftover allocation*, the strategy is to automate what functions *can* be automated, and allocate the remaining functions to the human.

In *economic allocation*, functions are divided between human and machine based on the maximization of some utility. Utility can be based on a single attribute, such as economic profit, or it can be the aggregation of many attributes (e.g., performance, reliability, safety), subjective evaluations, weights, etc. [91].

Why Have Prescriptive Allocation Methods Failed?

Each of the three general strategies seem reasonable, but researchers have long been trying to understand why they are rarely successful in real allocation tasks. There are differing opinions on this, but three reasons are frequently identified:

1. ***"Best" is not definable*** – Determining the "best" allocation assumes that "best" is definable, but this is rarely the case for the entire system. Jordan [67] argued that it is often wrong to *compare* humans and machines in the same terms (as Craik [25] suggested), and Price [116] claims that such context-dependent information is rarely available. Furthermore, when an overall metric for system "goodness" (such as a mathematical objective function) does not exist, an *a priori* optimal allocation strategy is not definable [139]. In fact, humans are often needed in decision systems precisely when explicit decision metrics are not available [125]. Lastly, defining what is best *a priori* (during system design) assumes knowledge of what is best during operation, but this is difficult when considering that designers have limited information about the specific operational situation [55].

2. ***It is difficult to model the human operator*** – When a human is part of a decision system, many complex issues arise in determining how automation is to support him or her because cognition is poorly understood. While cognitive models are valuable for predicting or understanding certain aspects of human behavior, their value in assisting function allocation has been limited. Price [115] claims that psychomotor and cognitive performances differ—the latter resisting analysis because cognitive tasks are not often overtly visible. Mental models for even slightly complex tasks remain elusive [137], and are often misused [129] and developed *ad hoc* [162].

3. ***<u>System</u> implications of allocation are complex*** – Allocation can have many system consequences that are difficult to understand. Even when a decision system can be

decomposed into multiple tasks for which a "best" can be individually determined, what is best for each *part* is not necessarily what is best for the *system*. This was recognized even by Fitts [43], who claimed that comparison lists are often misleading due to system effects. For example, automation may control a physical system more effectively, but humans that monitor automation can suffer a loss of situational awareness—an understanding of what is happening—which can be critical when human intervention is required. When humans get the "leftover" tasks, such as monitoring, they often become bored and feel alienated. These are just a few of the many system implications that may need to be considered in an allocation decision. Sheridan [138] provides a more thorough review of these issues.

The prescriptive methods—*comparison*, *leftover*, and *economic* allocation—can be useful for *guiding* function allocation once it is first determined where it makes sense to automate, since the allocation problem only arises when the human and machine can perform the same function [54]. However, among these functions allocation is not necessarily straightforward.

A proposed improvement to prescriptive allocation is to allow the human operator or automation to *dynamically allocate* functions [55]. This is common in many systems, such as cruise control in an automobile. However, this is often only a small part of a system in which the majority of functions need to be allocated during design. In addition, dynamic allocation often increases system and interface complexity [5].

Another common argument against prescriptive allocation is based on the principles of general design. Rouse & Cody [129] and Price [116] state that allocation can be systematized, but not to the extent of design prescriptions. Like other system engineering design processes, they claim that it is critical to retain human expertise and judgment in function allocation. Given that judgment is valuable for function allocation, the issue, then, is to determine how to improve function allocation in the absence or limited use of prescriptive methods.

1.3.2 Difficulties with Full Autonomy

It has been suggested, at times, to remove humans altogether from some decision systems. Proponents of fully automated decision systems argue that humans are unreliable and inefficient and that advances in technology makes this option feasible. It is informative to briefly review the past few decades of research in certain disciplines that have attempted to fully automate decision-making: Artificial Intelligence (AI), Operations Research (OR), and Management Science (MS).

- ***Evidence from Artificial Intelligence*** – Expert systems (one of the most popular early AI techniques), which attempt to capture the knowledge of experts in the form of rules, have fallen short of their initial claims: to surpass the abilities of humans [38], [162]. It is now often recognized that, in most practical settings, the primary value of expert systems is in *supporting* human decision-making [156], [167].
- ***Evidence from Operations Research*** – The early success of optimization algorithms for military purposes has led to applications in other complex systems, including civilian enterprises. The extension of traditional OR to these domains, such as social systems, has been generally unsuccessful [1], which has led to a new OR paradigm [126]. This paradigm incorporates the human as an active element in the decision system, in which traditional OR tools are still valuable components [22], [145], [166], [167].
- ***Evidence from Management Science*** – Managers, who are often faced with problems of maximizing profit under conditions of uncertainty, attempted to extend their standard analytical techniques to broader situations. It was discovered that the quality of decisions often degraded, which was explained as overly rational decision-making that intruded on the intuition of managers [38], [72] [131]. It is becoming more accepted that analytical tools are most valuable in management decisions when combined with expert judgment [58], [82]. This is evident in the designs of decision support systems for managers [70], [168].

The three fields discussed above—Artificial Intelligence, Operations Research, and Management Science—can provide lessons for the future design of human-automation decision systems. All have demonstrated success—particularly as computational resource limits have decreased—but have also been criticized for using algorithms inappropriately: as a *replacement* for humans. Today, it is recognized that in most reasonably complex decision systems, a human is an essential element.

1.3.3 Human Centered Automation

In recent years, engineers have supported the use of human-centered automation (HCA) for guiding system designs. HCA means “automation designed to work cooperatively with human operators in the pursuit of stated common objectives “ [15]. It is a design philosophy that recognizes (early in the design process) the unique attributes and requirements of humans as a functional component of the decision system [49].

HCA is valuable in that it directs design attention towards humans, but it has not significantly altered the manner in which functions are allocated. Sheridan identified problems associated with the interpretation of HCA, which can help to explain its limited use in the allocation of functions [136]. He claimed that at various times and in various contexts HCA is purported to mean:

- Allocating to the human the tasks best suited to the human, allocating to the automation the tasks best suited to it.
- Keeping the human operator in the decision and control loop.
- Maintaining the human operator as the final authority over the automation.
- Making the human operator's job easier, more enjoyable, or more satisfying through friendly automation.
- Empowering or enhancing the human operator to the greatest extent possible through automation.
- Generating trust in the automation by the human operator.
- Giving the operator computer-based advice about everything he or she might want to know.
- Engineering the automation to reduce human error and keep response variability to a minimum.
- Casting the operator in the role of supervisor of subordinate automatic control system(s).
- Achieving the best combination of human and automatic control, where best is defined by explicit system objectives.

These definitions of human-centered automation (some of which were discussed earlier as prescriptive methods) are not only problematic because they differ, but also because they are often undesirable and/or in conflict. For example, reducing human variability is in conflict with empowering the human. Similarly, there are cases when humans are not reliable or fast enough to be the final authority over automation. Other issues such as misplaced trust in automation and information overload are also discussed. One of Sheridan's conclusions was, while HCA is an appealing idea, the problem is that neither automation nor HCA is a *singular* idea.

In summary, given that humans are components in a decision system, even simple design philosophies do not lead to clear function allocation strategies.

1.3.4 Function Analysis

It is assumed that function allocation takes place at the higher levels of a design process (e.g., system architecture), before detailed decisions are made about the actual implementation. A common way to understand a system in terms of defined functions is through *function analysis*.

Function analysis is a design technique that allows designers to think abstractly about a potential design during the concept stage. It provides a way to express what the future product should do in terms of a set of functions that collectively satisfy the system function [16]. The main benefit of function analysis is that it helps designers concentrate on the "whats" before diving into the "hows" [86], [146]. In this thesis, *a function describes "what," and a process describes "how."* Generally, there are many processes that can satisfy a function, and hence, many ways to use humans and automation within a system.

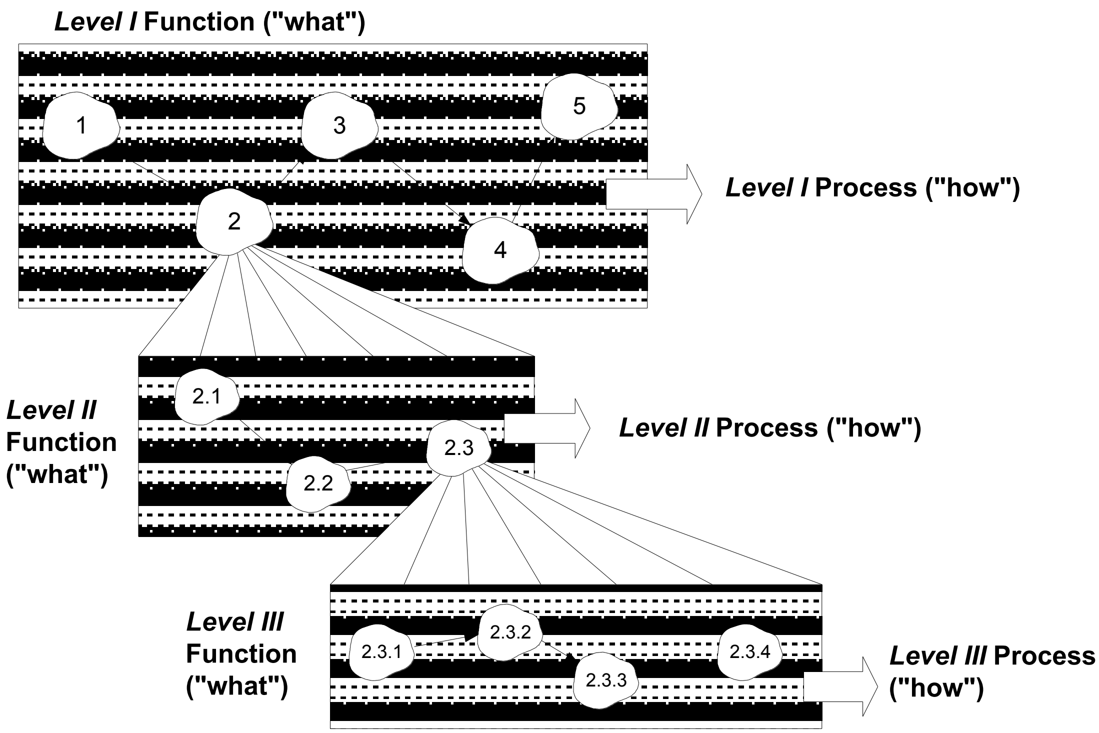


Figure 1-1 System representation for multiple levels of abstraction

For reasonably complex systems, function analysis may require multiple levels of abstraction. Figure 1-1 illustrates a functional hierarchy in which three levels are shown, the highest (*level I*) being the system function. Each level has a function ("what") and a process ("how"). The process may be described by "sub-functions" which can be further reduced, and the "hows" become "whats" as one moves down the hierarchy. Decomposition may continue to arbitrary levels of detail. However, at some level of detail the process cannot be further described (due to a lack of understanding) or is described so elementally that its further analysis is of little relevance. Such "elemental" processes are the conceptual building blocks of decision systems.

Function allocation takes place when a set of sub-functions has been defined that satisfies a system function, but their elemental process has not yet been determined. Therefore, function allocation is performed with a limited understanding of what the decision process will finally be.

1.4 THESIS OVERVIEW

1.4.1 Thesis Goal

The goal of this thesis is to provide insight into decision systems—particularly those which involve a mix of humans and automation. This insight may ultimately improve system designs by helping designers better-understand the implications of design choices, so that they use automation more appropriately within a system.

1.4.2 Approach

This thesis provides a novel way to analyze decision systems. The concept of a "semi-Structured" process is introduced, which recognizes both the well-defined "Structured" and ill-defined "Unstructured" parts of decision processes. The term "semi-Structure" describes how decisions are made, and therefore describes a *process* and not a *problem*. The semi-Structured framework can be used to qualitatively analyze systems by modeling their decision process, which distinguishes the parts of the system that are not completely prescribed prior to operation.

1.4.3 Chapter Descriptions

The body of this thesis has three main chapters. Chapter Two introduces the semi-Structured framework, while Chapters Three and Four apply the framework to example systems.

Chapter Two: The Semi-Structured Process

Chapter Two describes the semi-Structured framework for analyzing decision systems. After a semi-Structured process is defined, its properties and implications in decision systems are discussed. A significant portion of this chapter describes possible limitations of Structured processes, suggesting that there is value to Unstructured processes. Included in this chapter is a diagram notation that allows decision systems to be graphically modeled.

Chapter Three: Analysis of Example Decision Systems

Chapter Three uses the concepts developed in Chapter Two to understand and explain the designs of highly evolved decision systems. The purpose of Chapter Three is primarily to explore the utility of the semi-Structured framework for providing insight into decision systems.

The following example decision systems are analyzed in Chapter Three:

- *Temperature control*
- *Aircraft control*
- *Alerts and decision aids for aircraft collision avoidance*
- *Diagnostics and Procedures in aircraft and in medicine*
- *Multi-attribute decisions*
- *Engineering design*

Chapter 4: The Design of a Decision System for a Mobile Robot

Chapter Four is an exercise that illustrates how the semi-Structured framework might be applied to a design problem. Given a mission scenario for scouting a hostile urban environment, six sub-functions are defined for a robotic system. Based on these sub-functions, the design problem is to consider different operational decision processes that involve computers and a remotely located human operator. The goal is not to determine an "optimal" design, which is not definable, but to use the semi-Structured framework to understand the trades associated with design choices.

Chapter 5: Conclusions

CHAPTER TWO

THE SEMI-STRUCTURED PROCESS

2.1 INTRODUCTION

This chapter is primarily about decision processes. In the last chapter, it was mentioned that there often are many processes ("how") that can satisfy a specific function ("what"), and therefore many ways that humans and automation can interact in a decision system. Here, a paradigm is introduced for analyzing human-automation decision systems in terms of their processes.

2.1.1 Background: The Problem with "Problems"

Research in decision-making tends to focus on situations that are amenable to formal analysis. Many decisions, however, are believed to be in domains that are "ill-structured" [122] For these situations, it is often inappropriate to artificially structure a problem in order to apply formal decision methods, as in [164].

It can be difficult to determine, *a priori*, the conditions in which a problem can be solved solely by formal methods. Attempts have been made to classify problems based on formal criteria, but these have been unsuccessful. An example is Simon's analysis of "well-structured" and "ill-structured" problems [142]. The characteristics of a "well-structured" problem include:

- a definite criterion for testing any proposed solution can be defined
- a problem space exists that can represent the initial state, a goal state, and all potential transitions

- calculations are computationally practicable

Despite the apparent simplicity of the above characteristics, Simon made clear that a *formal* definition of a "well-structured" problem is impossible, such that there is no clear boundary between what is "well-structured" and "ill-structured." Reitman [123] describes problem structure as a continuum.

It is perhaps the vagueness of *problem* classification that deters people from formally determining when a function is fit for automation. However, problem attributes can still be valuable for making an informed allocation decision. It is often observed that humans are better than machines for "ill-structured" problems. These are the types of problems that are studied in "naturalistic" decision making, which are characterized by ambiguous information, incomplete and imperfect information, ill-defined and competing goals, high stakes, etc. [109]. Similarly, it is observed that automation tends to be better than humans for "well-structured" problems. However, many problems seem to fall in between the two extremes, making it more difficult to formally use problem characteristics to determine when a function should be automated.

2.1.2 Attributing "Structure" to a Process

Assuming that judgment is needed to help determine the appropriateness of automation, a more useful approach to decision system design is to focus on understanding the strategies used to make decisions, opposed to classifying the decision situation. Hence, this thesis uses "Structure" (vs. "structure") to describe *processes,* and not *problems*, as in [52] and [145]. Capitalization will be used to identify "Structure" accordingly. While the characteristics of a problem will be shown to influence the choice of process, there is no need here to formally classify problems in order to analyze decision systems for function allocation. That design task may be best left to the judgment of the analyst.

This thesis introduces the concept of a "semi-Structured" process. A semi-Structured process consists of both well-defined "Structured" processes, and ill-defined "Unstructured" processes, which interact in a known way. It is believed that many decision processes are semi-Structured, and that this concept can help to understand human-automation decision systems. For example, when a human acts as a supervisory controller, (e.g., giving commands to an aircraft autopilot and monitoring its behavior), the Structured process is allocated to automation (e.g., controlling the actuators), while the human's decision process as a supervisor may be poorly understood in certain situations, and hence Unstructured. Collectively, the semi-Structured decision process satisfies the system function (e.g., flying the aircraft). Semi-Structured decision processes are also observed in many other human-automation systems, as well as in human decisions and fully autonomous systems [90].

The semi-Structured framework provides a means for analyzing decision systems in terms of well-defined and ill-defined process components. Analysis can be performed somewhat independently of allocation. By first understanding a system decision process as semi-Structured, it may be possible to improve function allocation and hence the design of human-automation decision systems.

2.2 DECISION-MAKING AS A PROCESS

In the previous chapter, it was mentioned that the allocation of functions is often done during *design*, with limited knowledge of the process that would ultimately be used during *operation*. This section defines some characteristics of an operational decision process and its interaction with other systems.

2.2.1 Model of a Decision Process

During operation, a decision process transforms *information*, referred to as "inputs," and produces decision "outputs" (Figure 2-1). A decision process can be characterized by this input-output (I/O) transformation.

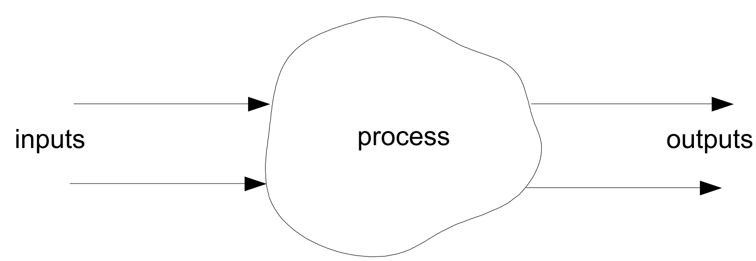


Figure 2-1 Generic notation for a "process"

In a decision system, processes are realized using humans and/or automation. In some cases, the decision process used by humans and automation are conceptually identical, despite differences in their respective internal machinery. For example, a human and a machine can use the same decision logic, obviously with different physical resources. In other cases, the decision *process* is also different. A process may be limited or constrained by the internal machinery of its (human or machine) host, but is not fully determined by it. For this reason, processes can be analyzed somewhat independently of their allocation.

2.2.2 Interaction with the Environment

A decision process is only one part of a larger system with which it interacts. During operation, a decision process is modeled to interact only through its inputs and outputs (I/O). This I/O represents information transfer to and from elements that are external to the process of interest—external in the sense that these elements can be represented as separate diagrammatic elements. External elements may be other sub-processes within the same decision system, but are often physically separate (such as a controlled plant). Collectively, these external elements are referred to as the "environment." Figure 2-2 illustrates a decision process interacting with an external element, which is one element within the environment (shaded area).

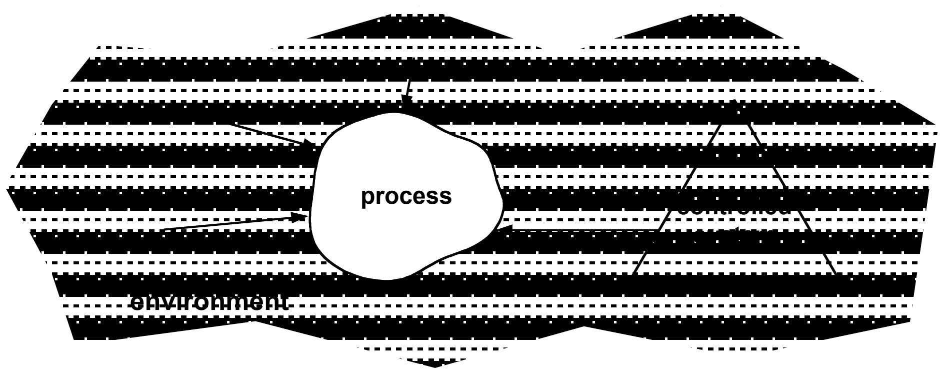


Figure 2-2 Process interaction with the environment

For the purpose of this research, it is assumed that decisions apply to controlling some system—intentionally altering its physical or informational state. Such "controlled systems" are represented in diagrams as triangles, as shown in Figure 2-2. A physical controlled system may be a vehicle, manufacturing plant, architectural design, etc. An informational controlled system may be an image, sensor data, computer model, database, etc. It is believed that the controlled system and the rest of the environment should be sufficiently understood in order for a decision system to satisfy its intended function.

Matching a Process with its Environment

In order for a decision process to satisfy a function, it needs to be appropriately "matched" with its operational environment. The interaction between a decision process and its environment is assumed to occur only through its I/O. Simon [141] identifies the I/O as the point of interaction between what he calls an "inner environment" (decision process) and an "outer environment." With this notation, he comments on the interaction of process and environment in satisfying a function:

> "…we often find quite different inner environments accomplishing identical or similar goals in identical or similar outer environments."

In other words, it is possible for different decision processes to satisfy the same function. Simon also writes:

> "If the inner environment is appropriate to the outer environment, or vice versa, the artifact will serve its intended purpose."

That is, a decision process (the "inner environment") can only satisfy a given function if it is "appropriate"[1] to the environment with which it interacts. When a decision process is prescribed prior to operation, the assumptions about the environment are critical to satisfying a function.

2.3 THE SEMI-STRUCTURED PROCESS

2.3.1 Definition

Decisions often involve two types of processes, defined as follows: a *Structured* process is a process that can be reduced to well-defined rules, while an *Unstructured* process is not reducible to well-defined rules. A process is "semi-Structured" when it contains both "Structured" and "Unstructured" sub-processes. This process distinction has many implications in decision system design, including the allocation of processes to humans and automation.

---

[1] The use of the word "appropriate" in Simon's quote suggests that formal criteria for satisfying a function may not be definable

Semi-Structured process – A system of Structured and Unstructured sub-processes

- **Structured process** – A process that can be reduced to well-defined rules
- **Unstructured process** – A process that *cannot* be reduced to well-defined rules

It is observed that semi-Structured processes are common in decisions. For example, automated processes are often governed by the formal language of computer code, and human decisions may be based on intuition or judgment that is poorly understood. In the context of function allocation, Structure does not imply allocation to automation, nor does Unstructure strictly imply allocation to humans. However, it will later be shown that a semi-Structured decomposition is an insightful foundation for making allocation decisions.

An inherent property of a semi-Structured process is that a decision system is *decomposable* into sub-processes whose interaction is understood. In fact, decomposability is a property that is fundamental to also defining the sub-processes. The primary difference between a Structured and Unstructured sub-process is the extent to which it is decomposable. Considering that a process can be characterized by its input-output transformation, further decomposition of a process implies further defining or constraining *how* this transformation occurs.

2.3.2 The Structured Process

A Structured process has been defined as a process that can be reduced to well-defined rules. A rule is a special type input-output transformation that can be represented or described. Furthermore, when a rule is "well-defined," the transformation can be *unambiguously* represented—for example, in the language of formal logic or mathematics. In this sense, a Structured process is completely decomposable: reducible to a set of primitive transformations whose further decomposition is unnecessary. In the diagrammatic notation of this thesis, a Structured process is represented as a rectangle (Figure 2-3).

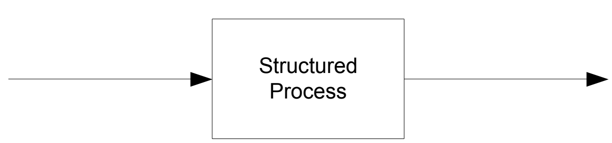


Figure 2-3 Diagrammatic notation for a Structured process

Some Implications of Representation

Inherent to the definition of a Structured process is that it is *understood* sufficiently to be *represented* with well-defined rules. This has important implications in its decision-making properties. For example, since a Structured process is completely defined prior to operation, it tends to be relatively inflexible. Furthermore, there are important implications in regards to its inputs and outputs. Recalling that a decision process should be analyzed in the context of the *environment* in which it will operate, a *Structured* process implies that the interaction with the environment can be explicitly represented. In other words, an implication of well-defined rules is that the decision process has an explicit representation of inputs and outputs, which requires a sufficient understanding of not only the rules, but also the environment with which they interact.

This last point can be further explained by recognizing that a Structured process is a symbolic process: one that maps input symbols to output symbols. It is not necessary that this process be a representation of any other system, such as a model of a physical system. A symbolic process is only a means for satisfying a function through its inputs and outputs. Just as a word is a symbol that exists only for its meaning, a Structured process is a symbolic process that exists only for its intended function. The concept of Structured process as a language is explained by Winograd [162]:

> "The very notion of 'symbol system' is inherently linguistic, and what we duplicate in our programs with their rules and propositions is really a form of verbal argument..."

Using this verbal analogy, it should be clearer to understand the importance of assumptions about the environment. Just as a word may have an intended meaning only in certain contexts, a process may satisfy a function only in certain environments. A Structured process operates on symbols *independently of its function*. Hence, inappropriate assumptions, made during design, can lead to unanticipated behavior [160].

Computer Code: A Test for Structure

The concept of treating decision processes as well-defined rules is founded in mathematics and formal logic: the principles behind calculating machines. In fact, this idea was extended as a model of cognition, as described by the *physical symbol systems* hypothesis [102]. For this research, computational metaphors are recognized as useful for describing rules.

While it is not necessary for a Structured process to be articulated in computer code, or implemented on machines, computer code serves as a sufficient condition for Structure:

> *A test for Structure is when a process can be reduced to a traditional computer algorithm.*

A computer algorithm is defined as "traditional" when it can be represented by production rules (in the form of IF…THEN statements) or mathematical functions. These rules are often in one of two forms: the *imperative* rule, and the *functional* rule [149]. The former views computation as steps or *actions*, while the latter emphasizes the computation of *values*. In either case, the rules are unambiguous and explicit.

Non-deterministic Issues

Well-defined rules are not restricted to deterministic operations. In fact, deterministic rules can be considered a special, limiting case of a more general probabilistic rule. When the output is a random variable, it is still generated by a well-defined rule because the process can be unambiguously represented. For example, if the process is a coin flip, there is no ambiguity in *how* the output is generated, only uncertainty in the *result*. Hence, a Structured process does not imply that the output can be predicted precisely from the input, but that the rules for generating the output are precisely known.

2.3.3 The Unstructured Process

By definition, an Unstructured process cannot be reduced to well-defined rules. Simply put, an Unstructured process is like a "black box": its inputs and outputs may be definable, but the process that governs their relationship is ill-defined. A process is Unstructured because it is not sufficiently understood to the extent that it can be represented with well-defined rules. In the diagrammatic notation of this thesis, an Unstructured process is represented by an oval (Figure 2-4).

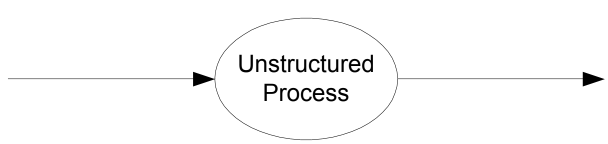


Figure 2-4 Diagrammatic otation for an Unstructured process

Since an Unstructured process is not necessarily constrained by well-defined rules, it may have properties that differ significantly from those of a Structured process. Some important implications of Unstructured processes include:

- *Inputs may not be completely definable* – Since an implication of rules is that the inputs are definable, Unstructured processes may not have definable inputs. Conversely, if the inputs cannot be clearly defined, this suggests that the process is Unstructured.
- *A priori optimization is not definable* – Since optimization implies the use of rules (e.g., maximizing an objective function), decisions made with an Unstructured process cannot be *a priori* optimal. One can argue that if an objective function is definable, then the rules for deciding are definable, and therefore the process can be Structured.
- *Increased Flexibility* – Since an Unstructured process cannot be fully decomposed, there are fewer known constraints. Therefore, an Unstructured process *may* be more flexible, adaptive and unpredictable than a Structured process, which is fully determined prior to operation.
- *Decision-making without Representation* – An Unstructured process may develop decision-making capabilities without any symbolic representation. Therefore, an Unstructured process is not limited by what can be explicitly *articulated*.

In summary, a seemingly simple process distinction based on well-defined rules has far-reaching consequences in decision-making. By relaxing the constraints on the extent to which a process is defined, an Unstructured process can be based on principles that are fundamentally different from those associated with Structured processes.

2.3.4 Structured vs. Unstructured: A Modeling Choice

Ambiguous Process Classifications

The Structured/Unstructured dichotomy forms a simple classification of processes that can be useful for understanding decision-making. Part of the value of the semi-Structured model of decision-making is its simplicity. However, a question such as “Is this Structured?” can arise during analysis if the classification is ambiguous.

In order to consider such questions, it is important to understand that Structure is not necessarily an objective property. Structure reflects the degree to which a process is explicitly understood, and is therefore, to a certain extent, subjective. The concept of semi-Structure illuminates the fact that decisions can be made with and without an explicit understanding of the underlying process. If a model of a process provides useful insight, then it is a valuable tool for understanding. Rather than asking "Is this Structured?" a more relevant question is "What can be learned from each (Structured or Unstructured) model?"

The Dynamic Property of Structure

Structure can also be considered *dynamic*. A process that is initially Unstructured can evolve towards Structured as the process is better understood. Hence, Structure can sometimes be "discovered" with sufficient analysis.

The discovery of Structure can be illustrated by assuming, for the time being, that every process is composed of well-defined parts that have yet to be uncovered. Like a complex machine, a decision process at first may be treated like a black box whose inner workings are poorly understood. After some analysis, it is often possible to identify some Structured component within the process, as well as its associated inputs and outputs. The result is a semi-Structured process with a defined sub-process interaction.

The dynamic property of Structure is relevant primarily over long periods. Since the evolution of processes generally moves in one direction—towards Structure—system designs tend to become more automated over multiple design generations. Structured processes may be considered the final evolutionary state of Unstructured processes, in which the mechanism for evolution is analysis. In the mean time, semi-Structured processes provide a way to distinguish those parts that are not yet explicitly understood.

2.4 STRUCTURE AS A DESIGN PRESCRIPTION

The difference between Structured and Unstructured processes has been defined primarily based on the extent to which it can be represented. A Structured process can be explicitly represented, such that its specification is complete and unambiguous. In the context of decision system design, the Structured/Unstructured process distinction is related to the distinction between the period of "design" and the (future) period of "operation."

Decision system designs are different from many engineering system designs because it is not necessary to *completely* specify the operational process—the means for satisfying a set of design objectives. Only the Structured parts of a decision process are those that are completely specified prior to operation, thereby eliminating these decision choices from the operator. Structure therefore can be viewed as constraints imposed on the operational decision process, which gives the designer explicit control over how these future decisions will be made.

Unstructured processes can be viewed as those parts of a system that a designer has deferred explicit decision control until operation. At first, this may seem irrational since it imposes a greater degree of uncertainty in *how* operational decisions will be made, which limits the degree to which a designer can exploit what is explicitly known prior to operation. In a sense, relinquishing control of the operational process limits the ability to "optimize" the system within a certain operational domain. However, Unstructured processes may be more appropriate because they are *not limited* by what is explicitly known prior to operation, which, among other things, may add flexibility and robustness.

2.5 DIAGRAMMATIC NOTATION

In this work, decision systems will be illustrated using a diagrammatic notation, shown in Figure 2-5. As mentioned, Structured processes are represented by a rectangle, Unstructured processes by an oval, and external systems by a triangle. Other symbols are also introduced. This notation helps to understand decision systems and the implications of allocation strategies. In particular, the notation allows for the explicit representation of the ill-defined parts of a decision process, which may otherwise be overlooked because they are not well-understood.

As mentioned earlier, it is assumed that in this work decisions apply to controlling some *external system* (represented by a triangle) in order to intentionally alter its physical or informational state. In the control paradigm, this system often represents a physical plant. A controlled external system may also be an information system such as a database.

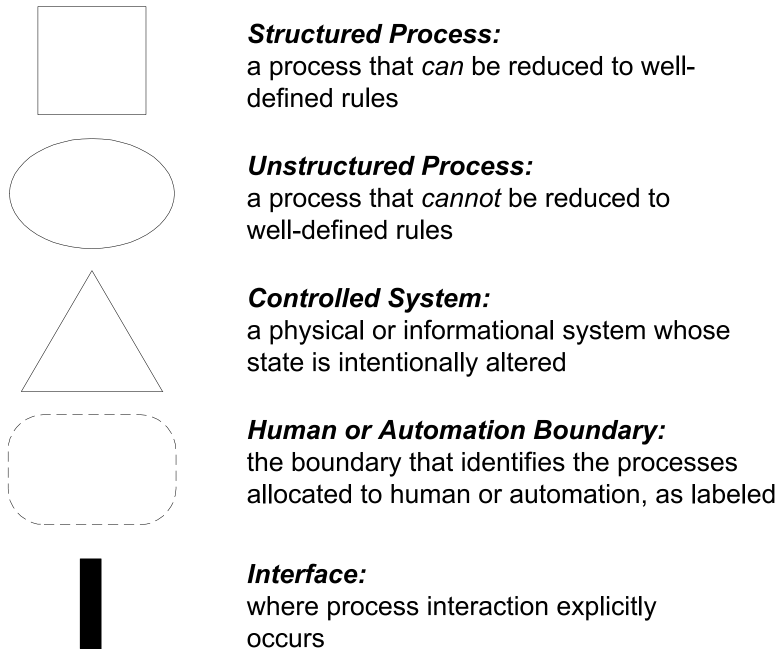


Figure 2-5 Diagrammatic notation for modeling decision processes

In order to identify which components are allocated to humans or automation, a dashed line is used as a *boundary*. *Interfaces* are implicit to communication between any of the decision processes and other elements in the environment. However, they are explicitly represented when there is value to identifying a means for communicating specific I/O. These conceptual interfaces tend to be particularly important when realized as physical human-automation interfaces.

Information flow between processes is represented by black and gray arrows (Figure 2-6). The distinction between well-defined information (black arrows) and ill-defined information (gray arrows) allows diagrams to capture this dimension of decision-making. Well-defined information often includes mathematical symbols, sensor measurements, or any descriptions that are unambiguous. Ill-defined information often includes words in natural language or any representation that has multiple or ambiguous meanings. The ambiguity of information is generally dependent on the context of its use.

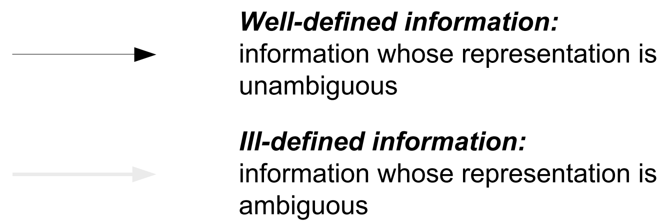


Figure 2-6 Diagrammatic notation for information: the inputs and outputs of a process

Although humans are not always able to unambiguously describe information, this may not be a hindrance since ambiguous information can sometimes be processed effectively for decision-making. However, ambiguous information affects Structured and Unstructured processes differently. In particular, a Structured process operates independently of ambiguity, so when the symbols are ill-defined, the decision process may not be effective. Since the allocation decision may be influenced by the ambiguity of information, and diagram representations can be important in open-ended design tasks [50], distinguishing the ill-defined components of a decision process can be valuable.

2.5.1 An Example Diagram: Supervisory Control

In order to illustrate how a typical decision system might be graphically represented, consider the generic human supervisory control system shown in Figure 2-7. In this model, commands are provided by a human's Unstructured decision process. The supervisory control decision is often given to the human because there are aspects of the decision that cannot be fully captured in a set of rules. However, human commands are processed by automation, which controls the plant using well-defined rules. This allocation strategy is typical of semi-Structured processes: there exists a portion of the system decision that is well-understood, which in this case is automated. Here, the flow of information is generally from left to right, which characterizes the *topology* of the semi-Structured decision process as *Unstructured into Structured*.

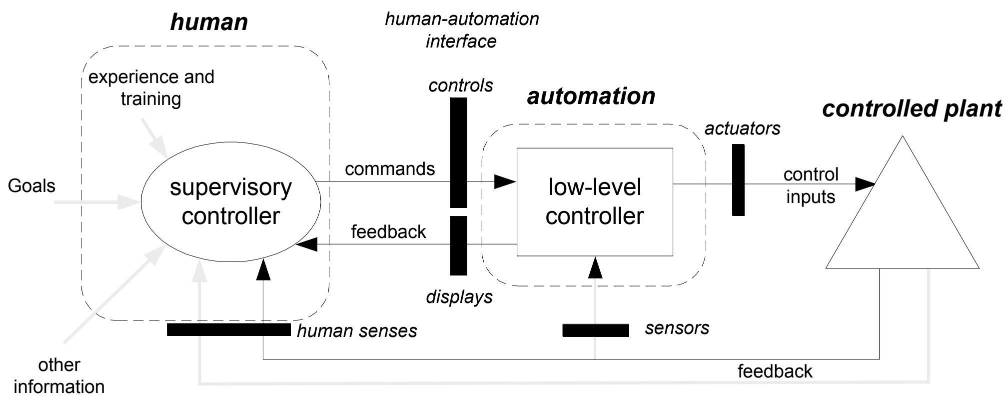


Figure 2-7 Generic supervisory control system

Structured and Unstructured Processes

The roles of the human supervisor and the automated low-level controller can be explained based on their respective processes. First, consider the Structured process. Its function is often to control the plant to the commanded states provided by the human. Through sensor measurements, it closes the loop on this state using well-defined rules, which are typically developed using feedback control theory. In contrast, the human is not merely a rule follower. The supervisory controller may use procedures for portions of his or her task, but is valuable for being able to resort to more complex, ill-defined processes. Humans are able to use experience effectively, adapt to changes, achieve ill-defined Goals, and serve as an effective monitor of automation. In addition, people are assigned responsibility, and are expected to use other resources, beyond standard procedures, to maintain system operation and safety.

By decomposing the decision system in this manner, both processes are exploited in a mutually constructive manner. Automation typically handles one or more well-defined tasks, usually at a high bandwidth, based on commands from the human. This allows humans to concentrate the ill-defined parts of the decision, requiring interaction with automation at less frequent intervals.

Supporting Information Requirements

Both well-defined and Ill-defined information is involved in the generic supervisory control decision system. It is important that the system is designed to support the information requirements of each process. The diagram in Figure 2-7 allows designers to understand these information components.

The information requirements for the Structured and Unstructured process differ. Note that automation is associated with well-defined information, which is often required for proper functioning. Note also that the human uses both well-defined and ill-defined information. While the system cannot be designed to prescribe *how* the human makes an Unstructured decision, the system can be designed to make the human's decision process *more informed*.

Well-defined Information

Well-defined information starts with commands from the human, which often represent controllable target states or goals for the automated controller. The computer translates this information, using well-defined rules, into outputs that control the plant through an actuator. Due to uncertainty in the plant dynamics, noise, etc., automation also uses well-defined feedback from the plant, as measured by one or more sensors. As Figure 2-7 shows, automation is often used to control *certain aspects* of a plant, characterized by well-defined states that are controllable and observable.

Ill-defined Information

In contrast, ill-defined information can play an important role in the human's Unstructured decision process. First, there is feedback from the controlled plant, which involves information that has both well-defined and ill-defined aspects. Whereas well-defined information may be a physical state that can often be easily measured with sensors or human senses, ambiguous information can be any other characteristic of the controlled plant that is observable but not easily describable. Both are observed by the human directly through his or her senses, which can provide additional and/or redundant information for monitoring.

The *Goals* represented in Figure 2-7 are shown as information coming from an external source: outside the *human* boundary. This is the generic representation, an example of which is receiving orders from a co-worker. Goals may also be internally generated, but are shown here as external to the Unstructured process so that their representation is explicit in the diagram. Furthermore, since goals associated with the purposeful behavior of humans can have many different interpretations [96], it will be capitalized (i.e., "Goals" vs. "goals") to distinguish it from the target state of automation, which is often explicitly provided by a human.

Since information from *experience and training* can affect Unstructured decision processes, it is also represented as an ill-defined input. Experience is also shown internal to the human, but external to the Unstructured process, so that the representation is explicit.

Finally, "*other information*" is a generic representation that includes any information that may be used by an Unstructured process during operation. Since the Unstructured process and its operational environment may not be well known *a priori*, "other information" covers unanticipated information, or information that is not explicitly prescribed as part of the decision process. This information *might* be well-defined during operation, but is not prescribed prior to operation.

The ill-defined information associated with the human's Unstructured decision process varies greatly in its type. Each can be critical to an operational decision, and collectively illustrate the different information which are often inaccessible by Structured processes.

Interfaces

Five types of interfaces are shown Figure 2-7. These include the sensor and actuator, which are interfaces between the automation and the plant. Human senses are also represented as an interface, in this case to identify information that is directly perceived from the controlled plant. As mentioned, an interface is always implied to exist at the boundary of the human, but is not always valuable to represent in diagrams.

A particularly important *interface* is that between the human and automation. This human-automation interface accommodates the well-defined controls from the human, and also provides feedback to the human through information displays. This feedback to the human is different from that sensed directly, since it is transformed to another representation that often provides added value. Certain information (and its representation) may be known to make the human's Unstructured decision more informed, and the human-automation interface is often the means by which this information is provided.

2.6 ALLOCATION TO HUMANS AND AUTOMATION

This section discusses various issues associated with the allocation of Structured and Unstructured sub-processes to humans and automation. The purpose is not to define which allocation is best, but rather to examine the ways in which Structured and Unstructured processes are often realized in decision systems, and to understand some implications of allocation decisions.

2.6.1 Humans and Unstructured Processes

Tacit Knowledge

From the previous discussion of Unstructured processes, their dominant role in human decision-making may not yet be apparent. It is generally accepted that many of the seemingly simple things people do—pattern recognition, judgment, language understanding, reasoning, etc.—are, in fact, poorly understood. In some sense, it may seem contradictory that good decisions can be made without understanding *how* they are made. However, this observed situation is often explained based on the type of knowledge stored in long-term memory, known as "tacit" knowledge [114].

Tacit knowledge is believed to be used in decisions in which the underlying logic cannot be verbalized, including some types of expert decision-making [72]. Together with "declarative" knowledge, which is used in Structured processes, this dichotomy forms one of the most accepted taxonomies of long-term memory [40]. It is believed that tacit knowledge cannot be clearly articulated, which leads to the difficulty in eliciting knowledge from experts in the development of "Expert Systems" [36], [38], [59], [65]. In particular, experts often have difficulty identifying the information that drives their decisions, as this information sometimes takes the form of subtle cues or complex patterns. Nevertheless, experts demonstrate that quality decisions can be made using Unstructured processes.

Unfamiliar Situations

Unstructured decision processes often apply to unfamiliar situations. When humans are faced with an unanticipated decision situation, they often need to formulate new goals and create novel decision strategies. In such cases, it is not possible to resort to successful procedures from the past, since these are formulated for specific situations. Human decision-making for unfamiliar tasks is classified as *knowledge-based behavior* in Rasmussen's skill-rules-knowledge taxonomy of human behavior [120]. This class of decision-making behavior is poorly understood.

Knowledge Transfer

Lastly, humans can *communicate* decision-making knowledge without an explicit representation (language). For example, tacit or non-verbal knowledge can be obtained through observations and practice, (although it may be *initially* obtained through language and rules [38]). Tacit knowledge-transfer has been demonstrated in learning situations, in which neither "teacher" nor "student" ever articulates their decision process. On the other hand, good teachers often excel at communication—articulating *how* to make decisions—and not necessarily performing the decision-making act themselves. The two different ways of transferring knowledge—with and without explicit articulation—has important implications communicating knowledge to other agents—human or machine—in a decision system.

2.6.2 Humans and Structured Processes

Structured processes are also common in human decision-making. Unlike Unstructured decision processes, the use of rules is often a figment of some degree of analysis, in which case humans are consciously aware of the logic underlying their decisions [32]. However, even when a decision is not self-analyzable, the decision process may be inferred to be Structured from the view of an observer.

Verbal Language as a Representation

A common (but not exclusive) medium of rule articulation is with verbal language. Verbal language provides a representation that can be used to capture declarative knowledge, opposed to tacit knowledge. Despite the fact that human language is largely imprecise [162], rules can sometimes be well-defined—especially within a limited context or domain. These special well-defined cases allow natural language to be sufficient for representing Structured decision processes.

Benefits of Structure

Humans find value in well-defined rules for various reasons. First, rules can be easy to *memorize* and recall [106]. For example, it is valuable to have a checklist before travelling, rather than develop a list from scratch. Rules can also lead to *repeatable* behavior in the appropriate environment. For example, there is value to saving a rule that is discovered through inefficient trial-and-error, assuming it can be applied successfully in future similar situations. As a social example, scientific theories often depend on a clear description of the experimental procedure in order to repeat the experiment for verification and acceptance within the scientific community.

Furthermore, rules allow knowledge to be transferred among people—to be communicated—without the need for direct experience. Just as a child may be verbally taught to "look both ways before crossing a street," rules allow less-experienced people to obtain knowledge from those with more experience. In fact, rules provide a means for *multiple* people to explicitly represent their collective wisdom. Well-defined rules are an unambiguous description of a decision process, providing a common representation for their designers to share knowledge, to modify or evolve as knowledge is accumulated over time, and ultimately to communicate this knowledge for others to use in the future. The benefits of this manner of knowledge-transfer are perhaps most apparent in societies and organizations, where rules or laws provide orderly, goal-oriented behavior at social scales. In the context of system design, Structured processes are a means for completely prescribing how operational decisions are to be made.

Robustness Issues in "Analytical" Thinking

When humans are executing Structured processes, their decision-making is sometimes classified as "analytical" (other adjectives include verbal, or logical) versus "perceptual" (nonverbal, intuitive) [130]. These two modes are often attributed to "left brain" and "right brain" thinking, respectively. Analytical and perceptual styles of decision-making have been shown to be influenced by the presentation of information (e.g., symbolic vs. perceptual), as well as the type of problem and the level of expertise of the decision-maker. While one cannot generalize so far as to say which mode is better (in fact, they are often recognized as complementary [106]) it appears that the decision "errors" in analytical thinking tend to be much more severe, while intuitive decisions are less precise but more robust [29], [53].

Standard Operating Procedures

Standard operating procedures are an important application of Structured processes in human decision-making. They are a set of experienced guides to behavior, sometimes executed without an understanding of the underlying rationale that led to the procedure [130]. For example, checklists provide a consistent way to ensure that certain criteria are met, such as the proper configuration of an aircraft prior to take-off [107]. Emergency procedures provide a way to make quick yet adequate decisions under time pressure [17]. Medical procedures provide society with more consistent medical treatments. Repair manuals allow inexperienced people to apply some of the knowledge of experts. When people use procedures, their behavior is *rule-*

based within Rasmussen's skill-rules-knowledge taxonomy of human behavior [120]. Procedures can be formulated in any number of ways, and may not even be justified as superior to other methods. However, they are deemed appropriate by one or more people, often after some degree of evolution in which the procedures are refined over time.

2.6.3 Automation and Structured Processes

The automation of traditional algorithms—those articulated explicitly in forms such as mathematical functions and production rules—has become ubiquitous. It is not necessary to elaborate on this, except to point out that the common use of automation is primarily a consequence of Structure. There may be value to the Structured process itself, but often there is separate or additional value to its *automation*, for reasons such as cost, repeatability, safety, and performance.

Given a Structured process, the primary difference between humans and machines are the resources available to implement the process. Craik [25] had stated that machines could replace humans in tasks where the human is understood as a machine—that is, for Structured processes. In such cases, when humans face sensing or cognitive "limits" in information processing (memory, attention, etc.), machines are generally able to implement the same Structured process with higher speed, and with greater precision and repeatability. Hence, while Structured processes often produce, for example, repeatable and predictable behavior within certain environments, these attributes can often be further exploited when the Structured process is *automated*.

2.6.4 Automation and Unstructured Processes

Automated Unstructured processes are not yet common in decision systems, but represent a growing class of decision-making algorithms that are fundamentally different from traditional algorithms. This difference lies not in its primitive logic, but in the conceptual level in which the algorithm is developed: with less emphasis on explicit representation [87].

The concept of an automated Unstructured process is based on the degree to which the operational decision process reflects the declarative knowledge of the designer. Essentially all computer code is ultimately composed of well-defined elements. However, some algorithms are not designed in a completely specified manner. Instead, knowledge is incorporated at a conceptual level that is different from the conceptual level of traditional code. The resulting operational decision process is not completely specified by the designer, but only *constrained*. For example, a designer may only specify how an algorithm learns or adapts. Under some situations, evolution between design and operation may produce an operational decision process that is essentially unknown, and therefore Unstructured. Since this can occur in varying degrees, the decision process modeling choice—Structured or Unstructured—remains a judgment of the analyst.

Neural Networks

In this thesis, "neural networks" are used as the main example of automated Unstructured decision processes. These are based on principles of connectionism versus symbolism, and are chosen here because they have attributes that make them particularly suitable for comparing to humans, such as the ability to generalize from experience. Neural networks offer a fundamentally different approach to automation, and provide *potential* value in many situations where humans are traditionally required. Other possible candidates include modern classes of adaptive, self-organizing, or emergent algorithms, including cellular automata [80], [94] (fuzzy logic is considered Structured because its classification logic is based on explicit rules).

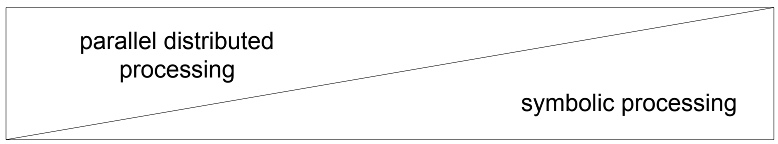
For this research, it is assumed that neural networks are of the "supervised learning" type. Unlike a traditional algorithm, in which a human is required to transform his or her knowledge into an explicit set of rules, a neural net creates—rather, strongly modifies—its rules using data from previous decisions. This data is selected by a human (supervisor). If successfully trained, a neural network should be able to operate on different inputs with the appropriate response. That is, it should be able to "generalize" from training situations to operational situations. The ability to generalize is dependent on both the internal organization of the neural network (neuron behavior, interconnection scheme, learning method) as well as the number and choice of input-output training sets [147].

Neural Networks: Decision-making without Representation

From the description of neural nets, it should be apparent that their decision-making capabilities can be obtained without a human's explicit understanding of their process. Just as humans can transfer tacit knowledge without symbolic language, a neural network can obtain the necessary "knowledge" to make decisions by observing the appropriate data.

Since neural networks are not constructed with an explicit understanding of the decision process, they tend to be have functional attributes that differ from more traditional algorithms.

Figure 2-8 illustrates a continuum of Artificial Intelligence computing methods, in which neural networks lie near the left end, and expert systems near the right (Barker [10] refers to the latter as "formal judgments"). Although the boundary between the two is unclear, the applications of each type of decision process has emerged over the years. For example, neural networks tend to be effective for pattern recognition and when the rule set is unclear, while expert systems are better when precise solutions are mandatory. Both types are often combined to form "hybrid systems" to exploit the advantages of each [10], [57], [90]. Hybrid computing systems are examples of fully automated semi-Structured processes.



- pattern recognition from raw data
- precise solutions are impossible
- inputs are noisy
- rule set is unclear, too large, or too dynamic
- sample decisions are available

- manipulation of abstract concepts
- precise solutions are mandatory
- inputs are accurate
- rule set is clear, and of manageable size
- sample decisions are difficult to produce

Figure 2-8 Continuum of Artificial Intelligence computing methods (adopted from Barker [10])

When the training process is from human decisions, the neural net can potentially capture both declarative (Structured) and tacit (Unstructured) aspects of the human's decisions. However, a neural net will not capture rule-based behavior with precision [83]. For example, it may generalize from an observed summation process that 2 + 2 = 4.03. Therefore, while the inferred Structured component of a human decision will not be precise (as the correct traditional algorithm would be), an advantage of neural networks is that they may be able to generalize to accommodate the portions of a process that are not understood in a declarative sense. In the context of decision system design, a system may be less limited by the declarative knowledge of the designers, reducing the amount of the operational decision process that has to be prescribed prior to operation.

Limits of Neural Networks

Again, it should be mentioned that neural networks are currently specialized processes that have been successful in limited situations. While they provide a means for an Unstructured process to be allocated to automation, neural networks are not trained with the breadth of information that humans observe with life experience. In fact, neural networks are essentially limited to operating in environments that provide at least a subset of the parameters with which they were trained. Section 2.7 discusses additional reasons why humans may be more appropriate for executing Unstructured processes.

2.6.5 Some Allocation Implications

The previous sections described how Structured and Unstructured processes are individually used in human and automated decisions. It is clear that both Structured and Unstructured processes can exist in human decision-making, and that both *can* also exist in automated decision-making. However, Unstructured processes are typically allocated to humans.

In the absence of prescriptive allocation strategies, it is particularly important that a designer understands the implications of allocation decisions. Some of these implications can be illustrated with a simple example.

Allocation Example

Consider two allocation decisions of the same semi-Structured process, shown in Figure 2-9. In Figure 2-9 (a), neither of the two sub-processes are automated, while in Figure 2-9 (b) one sub-process is allocated to automation. For this semi-Structured process topology, the decision to allocate sub-process *S* to automation first implies that a human-machine interface is required: a display to receive information from the Structured process. A conceptual interface is implied to exist between the two processes in (a) but this is not represented in the diagram.

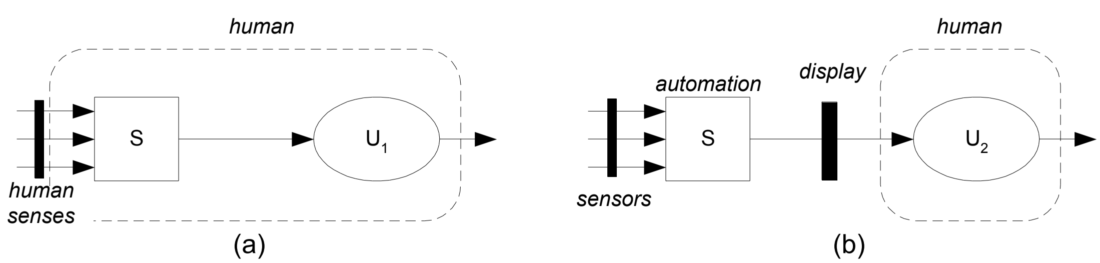


Figure 2-9 Information/interface implications of automating within a decision system

In addition to the human-machine interface, there is a difference in the information demands on the human between Figure 2-9 (a) and (b). In (a), the human accommodates *three* external inputs, perceived from his or her senses, while in (b), a *single* external input is required from the display. Similarly, automation has to process three external inputs, which may require sensors. The implications of automation on information flow and interfaces can be easily observed from such diagrams.

Adaptation of the Unstructured Process

When making allocation decisions, another important issue to consider is how Unstructured processes will be affected. It is difficult to comment on *how* an Unstructured process may depend on its allocation, since the process is not understood. However, it is important to understand *that* it can be affected by the way it is used in the decision system.

Unlike a Structured process, which remains conceptually unchanged despite its allocation to different to different hosts, an Unstructured process may vary between different humans, and between humans and machines. This is why the Unstructured process in Figure 2-10 (a) and (b) is distinguished by U_1 and U_2, respectively.

Furthermore, an Unstructured process may *adapt* in unpredictable ways to any change in the environment that is observable. Depending on the nature of the inputs, this may include the form and content of information, and hence the specific interface. Figure 2-10 illustrates how a different display—analog versus digital—does not affect the Structured process (as denoted by the unchanged label, "S"), but may affect the Unstructured process. The implication of this is that the same decision behavior cannot be expected when the environment changes. This can have both positive and negative effects. For instance, it may be desirable for a human operator to adapt to different interfaces. However, it may also be difficult for interface designers to understand and predict how design changes will affect decision-making.

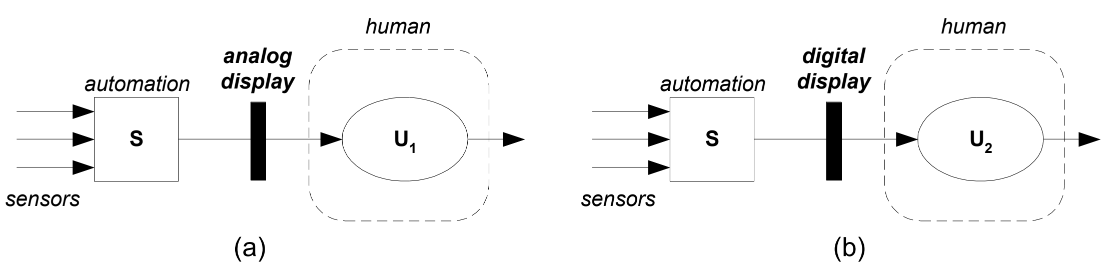


Figure 2-10 The effect of information form (display) on Unstructured processes (U_1 and U_2)

Similar issues arise when decision sub-functions are further decomposed or aggregated, (which may be motivated by allocation issues). Since a Structured process can be reduced to rules, the rule set can be further decomposed into smaller sets. This property allows for parts of a Structured process to be distributed among multiple humans and/or machines with (theoretically) no change to the process as a whole. The same does not generally hold for Unstructured processes, since the *process* itself is not understood. With Unstructured processes, it only makes sense to discuss *functional* decomposition, and not process decomposition.

2.6.6 Summary

The previous sections discussed various issues and implications associated with the allocation of Structured and Unstructured processes. In review, some of the important points from the way humans and automation individually use Structured and Unstructured processes are:

- Both Structured and Unstructured processes are common in human decision-making.
- It appears that humans add value particularly to Unstructured processes.
- Unstructured processes can sometimes be allocated to automation with algorithms such as neural networks, but these are currently restricted to special situations (such as when training data exists).
- Procedures are Structured processes that humans execute. These offer humans potential benefits related to memory/recall, repeatability, and knowledge transfer. In particular, procedures provide an explicit way to represent the collected wisdom of multiple people.
- For the same Structured process, automation often is able to execute rules with greater speed, precision, and reliability than humans. Hence, automation is recognized as a powerful tool for implementing Structured processes.
- Unstructured processes may allow tacit, nonverbal knowledge to be transferred.
- In order to function appropriately during *operation*, a neural network requires at least a subset of the parameters with which it was exposed to during *training*.

In addition, some implications of allocation were analyzed. The first example illustrated how the diagrammatic notation of Structured and Unstructured processes can provide some insight of how the information and interface requirements are affected by allocation. In particular, human-automation interaction implies the need for a human-automation interface.

The potentially adaptive property of Unstructured processes should also be considered when allocating processes. In particular, changes to the environment—including interfaces—can cause the Unstructured process to adapt such that the resulting behavior is difficult to predict.

In closing, what can be said about allocating functions to humans and machines? Given a functional requirement, a designer will likely have at least an estimate of whether the function can be adequately satisfied with a well-defined process. Even if such is the case, the allocation decision is a design decision, and therefore partly judgmental. This thesis does not provide design rules about what can or can't be automated; the allocation decision depends on the specific application. However, there are inherent properties of Structured processes that suggest when they may not be *appropriate*. This in turn may *suggest* the need for humans (or possibly neural networks) in the decision system.

2.7 REASONS WHY STRUCTURED PROCESSES MAY NOT BE APPROPRIATE

The purpose of this section is to examine possible reasons why a Structured process may not be *appropriate* in a decision system. While Structure/Unstructure design decision ultimately rests with the judgment of the system designer, the following sections provide an organized list of reasons to illuminate the implications of this design choice.

The majority of reasons why Structure may be inappropriate are primarily due to an insufficient understanding prior to operation, but it is sometimes possible to identify why this is the case. This section groups the list of reasons into four categories: *A* through *D*. The first category (*A*) addresses "internal" factors that cause an insufficient understanding—factors not associated with the environment. Category *B* is primarily associated with characteristics of the operational environment (e.g., process inputs and outputs, controlled systems), and is thus considered "external" factors. The third category (*C*) discusses humanistic requirements. Finally, Category *D* addresses practical issues associated with implementing Structured processes. An overview of the four categories is shown below.

Category A: Insufficient Understanding
- complexity
- miscellaneous
  - learning
  - analogical reasoning
  - lack of knowledge
	Category B: External Factors
	- ambiguity
	- insufficient information
	- uncertainty
	- adaptability
	- miscellaneous
	  - pattern recognition
	  - context

Category C: Humanistic Requirements
- subjective judgment
- moral judgment
- creativity
- responsibility
- miscellaneous
  - understanding goals
  - understanding intent
	Category D: Implementation Issues
	- information cost
	- processing resources
	- errors and robustness
	- design, verification, and maintenance

Category A: Insufficient Understanding

A Structured process may not be appropriate when humans do not sufficiently understand the situation. Given a function, it can be difficult for people to translate this into a sufficient set of well-defined rules, regardless of the characteristics of the operational environment (the focus of Category *B*). This category addresses *complexity*, and some miscellaneous reasons that lead to a lack of understanding.

2.7.1 Complexity

A Structured process may not be appropriate when it is required to be complex. Complexity is defined here as the difficulty in understanding input-output relationships due to an excessive number of interacting "parts," which in this context are sub-processes. A complex decision process is assumed necessary, given a complex decision task [8]. Even when it is possible to understand portions of a decision as simple, Structured sub-processes—perhaps with a high degree of certainty—complexity prevents a sufficient understanding of how the desired system behavior might be synthesized from these. As Figure 2-11 shows, large numbers of inputs and outputs may result from a complex process[2], but complexity is, fundamentally, an "internal" issue.

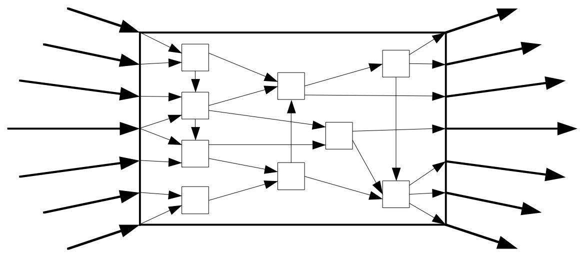


Figure 2-11 A representation of complexity in a Structured process

Reducing Complexity with Formal Methods

Three common ways to formally deal with complexity are through *elimination, aggregation,* and *functional decomposition*. Each method can effectively reduce the order of the decision process, respectively, by considering only dominant parts, selecting a coarser-grained model, or focusing on independent functions. The difficulty lies in applying such tools appropriately.

Elimination

One way to reduce the order of a decision process is through *elimination*, in which sub-processes and their I/O are selectively ignored (Figure 2-12). The result is a simpler process because the number of elements is reduced.

Eliminating elements is appropriate when done selectively, based on which are dominant or important in the system behavior. When considering these issues, a decision-maker needs to know which parts can be ignored while maintaining the desired functionality. This is essentially what formal modeling typically accomplishes: an abstraction of a complex physical system in which the dominant parameters are represented. However, when a process is complex, it may not be understood sufficiently to selectively eliminate elements.

---

[2] Reducing the order of information assumes that information is decomposable, and not "holistic."

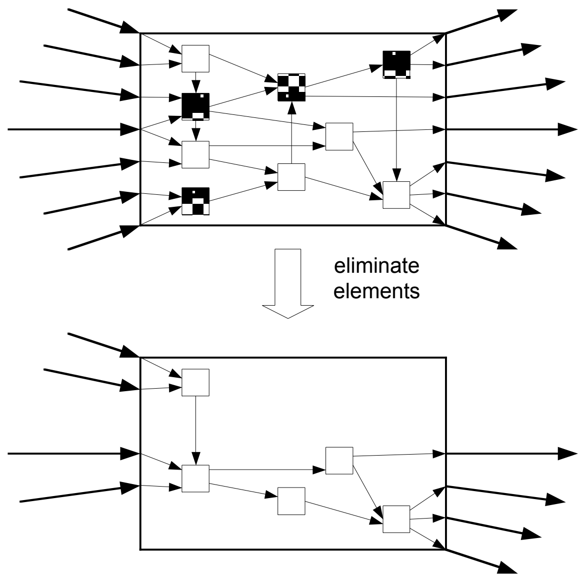


Figure 2-12 Reducing complexity through sub-process "elimination"

Aggregation

Another way to reduce the order of a decision process is through *aggregation*—the combination of constituent parts into a whole (Figure 2-13). In the context of decision processes, aggregation results in fewer, less-detailed sub-processes, but does so without the explicit removal of details. Rather, details are hidden implicitly by using higher levels of abstraction. One implication is that I/O is often more general, implicitly expressing the information it replaces. This is shown by the bold arrows in Figure 2-13.

Aggregating elements requires that the appropriate level of abstraction be chosen to capture the desired behavior while still achieving a reasonable reduction in complexity. For example, it can be appropriate to consider only the speed of an automobile, opposed to engine torque, transmission loss, wheel slip, road grade, aerodynamics, and wind—these are all factors which affect speed, but are not fully determined by speed. However, it may be necessary to consider these details for certain tasks. When a process is complex, it is difficult not only to understand the relationships between the detailed and general representations, and the conditions in which aggregation is appropriate for the decision.

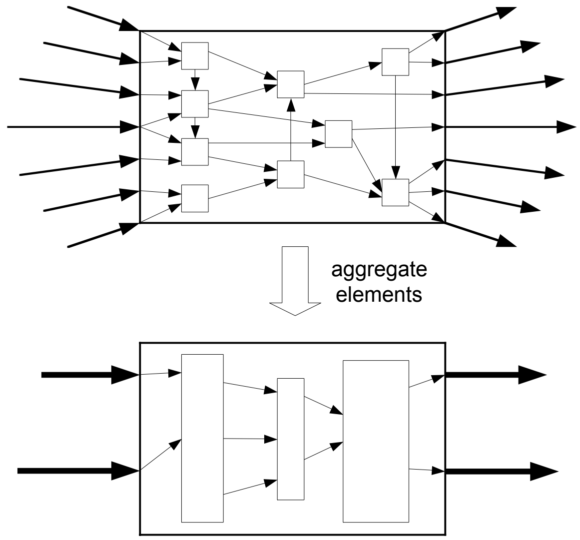


Figure 2-13 Reducing complexity through sub-process aggregation. The bold arrows represent information at a higher level of abstraction.

Functional Decomposition

A third method of dealing with complexity—functional decomposition[3]—is a "top down" approach that results in simpler sub-functions. Although this approach implies *process* decomposition, it is conceptually different from simply decomposing a complex process without first generating new sub-functions. The value of this approach is primarily that the reduced complexity of the sub-functions allows a simpler sub-process. However, the decomposition into sub-functions requires that the wholeness of the task is not sacrificed [85].

Three common functional decomposition strategies are serial, parallel, and hierarchical decomposition (Figure 2-14). A *serial* decomposition implies that the output of one function becomes an input to another. An example is an alerting system, in which raw information is processed to calculate relevant states that are then fed into alerting logic. A *parallel* decomposition may be the steering and throttle controls of an automobile, neither of which outputs are necessarily required by the other function. Hierarchical decompositions consist of serial and parallel functions, but these are arranged in a way that has discrete "levels" which are

[3] Functional decomposition was discussed earlier as an assumed part of *decision system design*. Here, it is discussed in the context of reducing complexity for operational decisions.

defined by their interaction. When modeled after human organizations, the functions in a single level are the means for achieving the functions in the level above [92].

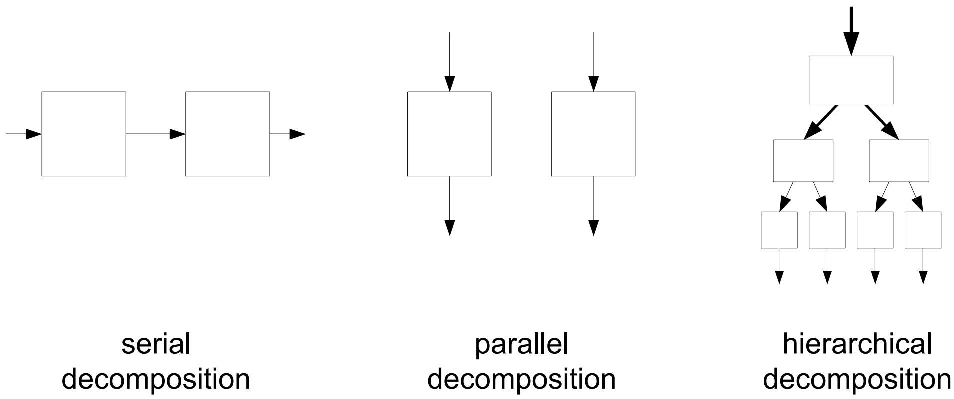


Figure 2-14 Three common types of functional decomposition

Reducing complexity through functional decomposition can be difficult without a deep understanding of the system goals. First, the assumption of truly independent sub-functions can be inappropriate, particularly because each sub-function is "selfish" or "myopic"—it makes decisions independently of other functions, and does not consider the system goals in its decision. It further requires that the collection of sub-functions can be evaluated to satisfy the system function, which itself an ill-defined task when the function is not explicit.

Advantages of Unstructured Processes

Unstructured processes may be able to accommodate situations that are considered complex for Structured processes, because the process does not have to be explicitly understood. "Complexity" itself may not defined in the same way as with a Structured process. Process elements may not be interpreted as separate components, but as a holistic pattern. By viewing a process as irreducible, Unstructured decision processes may not suffer from complexity. Furthermore, experience can provide techniques to effectively circumvent complexity.

Humans are known for their ability to handle complexity. Cognitive psychologists believe that humans deal with complexity, in part, through three techniques: pattern recognition, heuristics, and abstraction hierarchies.

First, it is believed that, with experience, humans are able to reduce *patterns* of otherwise complex information into simple "chunks" that can be easily recognized [140]. While humans appear to be limited to about seven chunks of information in short-term memory [93], the size of these chunks increase with experience within a domain. Experts are believed to have an intuition of expected patterns of information, and to detect anomalies in observed patterns based on this expectation. Hence, humans may not be aware of the large amount of information that is present, but perhaps only the small amount of information that is *not* present but expected. This is believed to be one reason why experts are to be less likely to fall victim to information overload [72]. Second, humans find means to reduce cognitive load by using biases or *heuristics* that can simplify the problem [130]. Heuristics are speculative strategies that are often adequate, but not *guaranteed* to work (although humans tend to use them effectively). Third, abstraction hierarchies allow humans to think in terms of different levels within a hierarchical mental representation. Details can be avoided, if desired, by crossing to higher levels of abstraction, in which representations are more general [120]. These simplifying techniques are perhaps necessary considering that human rationality is bounded [141].

Even when a situation is decomposable, humans are typically required to implement the formal simplification techniques discussed earlier. In elimination, humans identify the important or dominant parameters for modeling. In aggregation, humans choose the appropriate level of detail. In functional decomposition, humans understand the concept of function, and how the wholeness of the task can be preserved with the appropriate decomposition. Each of these methods seems to require a deep understanding of the underlying situation, based on learning and experience, and are therefore difficult to perform using rules. That is, humans appear to be valuable for their ability to apply formal strategies for reducing complexity.

Neural networks are also known for their ability to generalize correctly from complex sets of data [147]. When faced with a seemingly complex situation, a neural network may be trained without the need for humans to ever explicitly understand the process. This can lead to the *discovery* of data patterns, particularly in situations where perception cannot be exploited, and/or short-term memory becomes a bottleneck (e.g., high dimensional data analysis). Unlike humans, neural networks may not be able to access metaphors and analogies. In addition, it is important that their internal connectivity is properly matched (through design) to the complexity of the decision task. Nevertheless, neural networks have clearly demonstrated an ability to accommodate complexity—usually much faster than humans.

2.7.2 Miscellaneous Factors for Insufficient Understanding

Learning

A Structured process may not be appropriate when it needs to learn how to improve over time and experience. The mechanism for human learning is complex, as it involves an organization of perceived information as knowledge in long-term memory [40]. Experience provides data and information, but processes are required to use these effectively to store and recall *knowledge*, and to access this knowledge constructively for learning. Traditional learning algorithms (as in adaptive controllers) are limited in their ability, since this typically involves the explicit language of parametric models. In contrast, neural networks can learn by example—with little emphasis on representation—and may provide greater potential for incorporating experience effectively in decisions.

Analogical Reasoning

A Structured process may be inappropriate when it needs to refer to stored information for analogical reasoning. This ability allows a decision-maker to use previous decisions by interpreting the current decision situation as analogous to it in some way [68]. Analogues are useful for generating expectancies, solving problems, and making predictions when there are many unknown factors [72]. The hardest part is finding a good analogy, such that a previous decision can be identified and used in the appropriate way. While not inherently humanistic, analogical reasoning appears to be a difficult to accomplish with traditional algorithms as well as neural networks.

Lack of Knowledge

It is difficult to articulate a reasonable set of rules for decision-making when there is insufficient knowledge in a particular domain. This is a general point, but it is important to separate this from other issues, such as complexity. If one does not understand the fundamentals of a situation—the relevant issues, constraints, interactions, allowable modifications, etc.—there is no reasonable basis for a decision, even if the goals are clear. For instance, a child cannot be expected to perform surgery, let alone articulate how this should occur. A lack of knowledge can be identified as a general reason for not being able to articulate a set of decision rules, and can often serve as a broad catchall category after considering the other categories listed here.

Category B: External Factors

A Structured process may not be appropriate because of "external" factors. In contrast to the previous section, these are issues associated with the operational environment: inputs, outputs, and controlled systems. From what is known about the environment, it may be difficult to articulate a set of rules to satisfy a given function within that environment. In this category, the following reasons underlying the difficulty with rules are discussed:

- *Ambiguity*
- *Insufficient Information*
- *Uncertainty*
- *Adaptability*
- *Miscellaneous*

The brief explanation of why a Structured process may not be appropriate is because humans do not explicitly understand how to articulate the rules. The above list, which is considered in detail in the following sections, explains *why* this is often the case.

2.7.3 Ambiguity

A Structured process may not be appropriate because of ambiguity. This can arise in many ways—any time language or representation is involved [133]—but is examined here as three types of representations: ill-defined goals and ill-defined inputs and outputs (I/O). These ambiguous components are illustrated in Figure 2-15 as gray arrows.

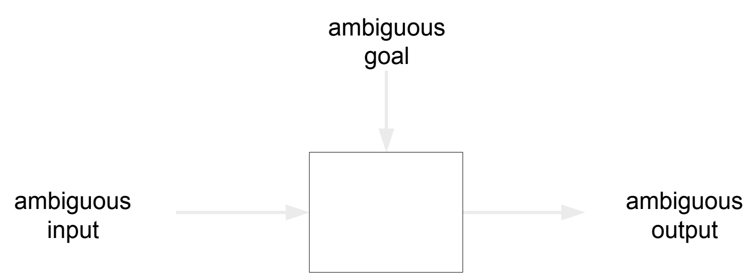


Figure 2-15 Ambiguous inputs and outputs (I/O)

Ambiguity can be particularly problematic with Structured processes because rules manipulate symbols independent of their meaning [33], [162]. When these rules are blindly executed by an agent (assumed different from the designer), the ability of the process to satisfy its intended function may be compromised. It is important to understand that ambiguity is an *interpretation* issue, which implies that a description of rules and/or goals is only ambiguous to *humans*—such as designers, analysts, or operational decision-makers. Gray arrows in the semi-Structured process diagrams can be associated with Structured processes, but are a tool for the human who interprets these diagrams.

Ambiguous Goals

Ambiguous goals are the foundation of ill-defined decision problems. It is not always possible to define a problem space in which the state of the controlled system can be explicitly defined. However, Structured search and optimization methods generally require that there exists a formal way to test if a proposed decision alternative is acceptable—which, according to Minsky is the definition of a "well-defined problem" [97]. Often, decision situations are characterized by ill-defined goals and open constraints [109], [123]. In particular, decisions that involve multiple considerations frequently do not have definable trades among these criteria, and cannot be aggregated into an overall scalar metric. In fact, the "state" of the controlled system can involve subjective assessments that cannot be articulated prior to the decision, but often can be easily assessed during operation (i.e., you know it when you see it).

As an example, the function of a vehicle guidance system may be defined as "minimize fuel and time." This may initially seem reasonable, but this goal can be ill-defined since minimizing fuel and minimizing time are often conflicting requirements. It may be possible to minimize fuel for a specified travel time, to minimize travel time for a specified fuel consumption, or to minimize some weighted combination of the two parameters. But without such qualifications, there is no clear basis for decision-making, and judgments may be required to resolve this ambiguity.

It should be mentioned that a Structured process does not require that a goal be explicitly defined, as in the "well-defined problem" described by Minsky [97]. Goals are not always an explicit input to a Structured process, but may be an implicit property of the rules, as determined prior to operation. This is frequently the case with standard operating procedures, for example. However, in many operational environments—particularly those that are not well modeled—an explicit set of rules may not be appropriate when the decision goals are ambiguous.

Ambiguous Inputs and Outputs

Ambiguity can also affect the function "bottom up" through the definition of the inputs and outputs (I/O). Even when the function is well-defined, it may not be achieved when the process interacts with the environment based on I/O that are not well-defined in a certain operational context. Since a Structured process is a symbolic process, it is important to understand the difference between I/O symbols and their meaning [162]. The two are sometimes distinguished by "data" versus "information," or by the capitalization of the "i" in "information" [33].

While it is typically possible to define the low-level sensory data inputs to a decision process (e.g., patterns of light), the definition of useful information from this data can be vague. As mentioned earlier, an indication of an Unstructured decision process is when the inputs cannot be clearly defined. Furthermore, the actions for altering the state of a controlled system—the decision outputs—may be easy to do, but not to describe. An example of ambiguous inputs and outputs is in medical decision-making (Figure 2-16). For instance, patient complaints of symptoms are often vague and descriptions of treatment can be difficult to describe, such as in surgery.

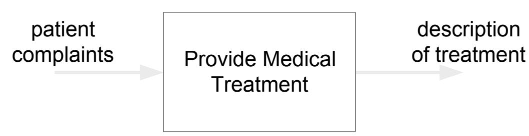


Figure 2-16 Ambiguous inputs and outputs in a medical decision

A critical factor in the interpretation of inputs and outputs is the *environment* in which the Structured process operates. Information that is well-defined in one environment can be ambiguous in others. Figure 2-17 illustrates this issue with a rule-based computer guidance and navigation aid for an automobile. If a rule states for a car to "turn left" at the next intersection, this interpretation is clear when the streets are arranged in a rectangular grid (Figure 2-17 (a)). However, the instruction can be ambiguous in other situations, as is shown in Figure 2-17 (b). In this example, there is ambiguity in the decision *output*, as represented by a gray arrow. Typically, though, ambiguity is an *input* issue, since it is more difficult to understand the information used in a decision.

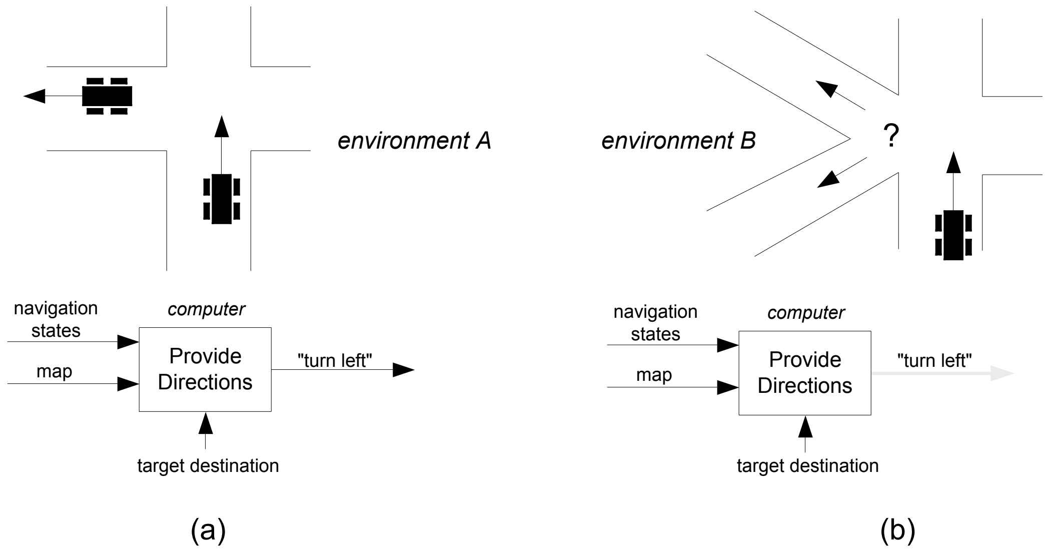


Figure 2-17 Automobile guidance: an example how the environment affects ambiguity

Advantages of Unstructured Processes

Ambiguity is a condition that arises due to representation, and therefore may not be problematic with Unstructured processes. Humans frequently deal with language imprecisions (such as in verbal communication), but are able to resort to ill-defined strategies to resolve ambiguity when it arises. These strategies should not be confused with "fuzzy logic," which provide a precise, rule-based way of accommodating the imprecise way that humans describe things, thereby *circumventing* ambiguity. In contrast, humans handle language imprecision using processes that are not well understood. For example, in the car guidance example a human can use context and common sense—a more complex "set of rules"—to resolve which street on which to turn.

Humans may also handle goal ambiguity, since they often understand the *implied* goals when these are not made explicit [123]. Humans share a set of common Goals and values that can be inferred for guiding operational decisions (the capital "G" denotes the humanistic use of "Goals"). For example, a pilot whose function is to "minimize fuel and time" may be able to select a reasonable travel time from which to minimize fuel, based on how the passengers or the airline values time. Common Goals are also important for safety since it is assumed that humans have a survival instinct that drives them to access whatever resources possible—beyond any predefined rules—in order to stay alive. When humans make decisions, such goals do not necessarily have to be made explicit.

Neural networks also may not suffer from ambiguous I/O. It may be trained with information, or patterns of information, that are humans cannot explicitly define. In image recognition, for example, the attributes of a facial image do not have to be described in terms of decomposed facial features (although it may seem that these attributes are used in the decision process). Visual patterns that are difficult to represent, or seemingly hidden, can still be detected and correlated with a neural network's Unstructured process.

In addition, neural networks may not be affected by goal ambiguity. For example, a neural network that is trained with input-output information from a human decision-maker does not require that his or her Goals be articulated. It is not suggested that a neural network construct its own internal goals (although it may be designed to do so). However, training data from human Goal-driven decisions can be used by neural networks to produce operational decisions that *appear* to have been driven by the same Goals.

2.7.4 Insufficient Information

A Structured process may be inappropriate when there is an insufficient set of inputs or outputs (I/O). Since rules require certain inputs, a Structured process cannot operate if these are not provided during operation—such as from failures in sensors or communication. Furthermore, insufficient I/O may be a condition that is known *prior* to operation, in which case rules cannot be articulated due to constraints in the operational environment—such as from imposed hardware. Both situations are illustrated in Figure 2-18.

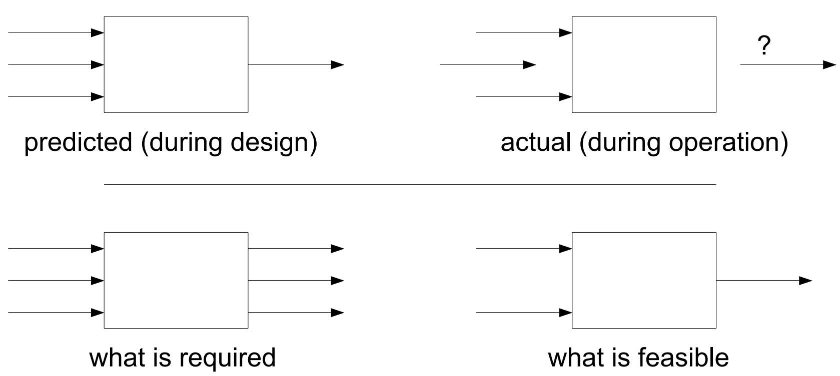


Figure 2-18 Insufficient information

The I/O requirements of a decision process can be dynamic or static. Dynamic systems may require a minimum communication rate or bandwidth. For example, in the controls domain, a controlled system may have internal dynamics that require a high-bandwidth controller for stability. This is likely to impose requirements on feedback information that has to be observed and processed at a minimum rate. When the information rate is too slow, the decision is considered under-informed in the dynamic sense.

Insufficient information in the static sense (when temporal characteristics are not problematic) can also be a limiting factor with rule-based decisions. If the required inputs are not present during operation, a rule is ineffective. For example, a rule such as, "IF (temperature exceeds limit) THEN (deactivate heater)" cannot operate if temperature information is not provided. Hence, a Structured process not only needs to be sufficiently understood in order to be appropriate, but the operational environment must also provide sufficient information.

Even when the operational environment is well-understood, it may be difficult to articulate a set of rules based on the available inputs and outputs. For example, in control theory a system that is theoretically "observable" and "controllable" may be difficult to control with a Structured process because the inputs and outputs *appear* insufficient to the analyst. In this case, it may be possible to articulate a Structured process for other inputs and outputs, but these may be impractical due to implementation issues. Hence, a situation that appears to be inherently limited by insufficient I/O (as would actually be the case for an unobservable/uncontrollable system), may only be due to an insufficient understanding of how to articulate a process based on a certain I/O set.

Advantages of Unstructured Processes

Unstructured processes may not be limited by the same conditions of insufficient information, since "insufficient" is defined based on what can be explicitly understood. Both humans and neural networks demonstrate abilities to make adequate decisions under such conditions. The "under-informed" case (insufficient *inputs*) is common since the decisions are often limited by the available information.

While rules cannot function with insufficient information (by definition), humans are good at "making do" with incomplete information [130]. Humans can make up decisions when rules become "stuck." It is believed that experience allows people to utilize tools such as intuition and analogical reasoning (solving a current problem by relating it to another problem that is reasoned to be analogous) in order to make decisions under conditions that are insufficient for rules. Analogues can be helpful when information is good quality, but there is not enough to apply a more rigorous analysis [72]. Humans are known to use context and experience to understand the "big picture" that may be associated with the definable information accessible by rules (such as recognizing a box from the view of only three of its sides). This is particularly the case with experts. Furthermore, there often exists useful information that is not well-defined—in the form of fine patterns or subtle cues—which allow humans to see the "invisible." In short, humans can often make decisions in under-informed situations, based on processes that are ill-defined.

Neural networks, like humans, are also good at tolerating missing information [83]. They can often find trends, extrapolate, and fill in states—usually faster than humans. However, neural networks are limited to operating on the same type of information with which it was trained, which may be deep in a particular domain, but not nearly as broad as human experience.

2.7.5 Uncertainty

A Structured process may not be appropriate for some types of uncertainty. Uncertainty relates to the possible knowledge about a situation or over time to any possible outcomes that derive from actions [166]. The deductive input-output logic of formal rules can produce erroneous outputs if the *inputs* are not known with sufficient certainty, even when uncertainty can be formally characterized. In addition to inputs, uncertainty applies to understanding the behavior of a controlled system, such that *consequences* of decisions are uncertain (e.g., financial investments). In this case, the results of a given action are not known—including probabilities (the likelihood of an event) and possibilities (an understanding of all possible events). While formal algorithmic tools exist to handle certain types of uncertainty in decision-making, they require a specific formulation that may not be appropriate for the problem.

Problems with Classical Decision Theory

Structured decision processes often use probabilistic models and utilities to formally handle uncertainty. These methods, which were developed for applications in economics, provide a quantitative basis for decision-making. A popular framework is classical decision theory, which prescribes how "rational" decisions should be made [35], [46], [71].

In classical decision theory, it is assumed that a plurality of decision consequences is known (C_1, C_2, …, C_m), and that each possible consequence has an associated probability (p_1, p_2, …, p_m). When a lottery (a probabilistic trial characterized by a mutually exclusive, collectively exhaustive set of consequences) is presented to a decision-maker, classical decision theory provides a basis for how a rational person *should* decide in the presence of uncertainty. An example decision situation is shown in Figure 2-19, in which two binary lotteries are presented.

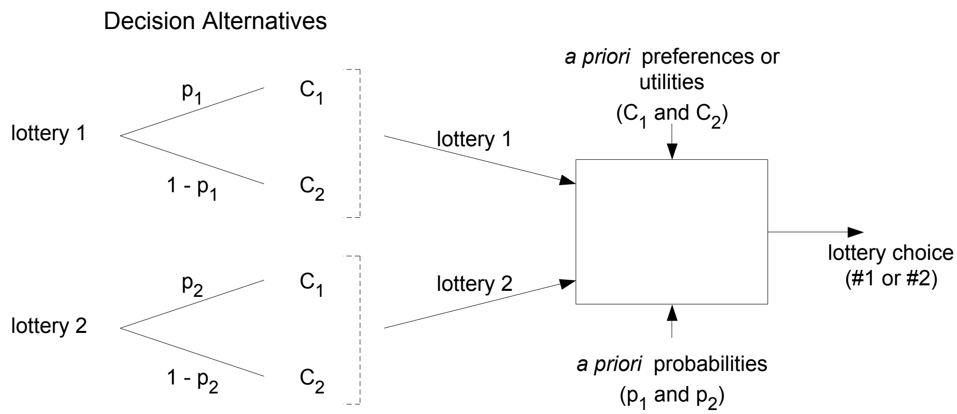


Figure 2-19 Binary lotteries: an example of a formal decision process with uncertainty

Figure 2-19 illustrates that classical decision theory provides an explicit form to model decision problems. The Structured decision-making logic is prescribed by the axioms of rational behavior, first introduced by Von Neuman and Morgenstern [153], which includes:

- Preferences and probabilities exist, and can be quantified.
- Preferences are transitive (if A>B and B>C then A>C, where ">" denotes "is preferred to").

These and other axioms lead to utilitarian decisions, in which rational decisions are those decisions that maximize the expected value of utility.

There are many reasons why classical decision theory has been criticized [7], [12], [60], [98], [111]. First, a complete set of possible decision options and their consequences is not always known. In most decisions, choices are not made from a fixed set of alternatives; they must be formulated or designed [143]. Second, probabilities are often unknown, or are difficult to estimate [12]. This problem is often seemingly circumvented by resorting to models that simplify analysis (such as Gaussian distributions) but are otherwise unjustified. Third, actual preference behavior has been shown to violate the existence of transitivity and utility [75], [60]. Preference information may be inappropriately transformed, and incommensurate information aggregated to a common scale in order to apply such decision rules. Decision situations can be sufficiently complex such that it is impractical to evaluate the utility for each outcome [130]. Fourth, the concept of utility has been shown to often be a poor representation of decision-making at the extreme values of utility and probability (near 0 or 1) [31]. In short, classical decision methods are normative, and often are not accurate descriptive models of human decisions. In fact, it has been shown that that such methods can degrade human decisions by eliminating intuition [36], [53], [73].

Advantages of Unstructured Processes

Humans appear to be capable of dealing with uncertainty. Investors make important financial decisions in the face of uncertain markets. Physicians make health decisions based on uncertain verbal information, or imperfect test data. Marketing strategists determine opportunities based on statistical sampling. In the military, potential targets have to be identified as friend or foe based on imperfect information. In these situations, people may not understand the situation in a way that allows formal modeling. Instead, they are believed to rely on judgment and intuition, in which risk is accounted for using ill-defined processes.

Neural networks also tend to be robust to uncertainty, as demonstrated by their tolerance to imperfect, noisy information [83], [90, [147]. In contrast, rules are precise and tend to be sensitive to imperfections in information. Neural nets may not offer precision, and (like humans) are not known to be good at tasks that require precise calculations or repeatability. However, their tolerance to uncertainty often makes them robust. In fact, noise is often introduced to training sets in order to improve robustness. Hence, training often provides the necessary information to allow neural networks to accommodate uncertainty without formal rules.

2.7.6 Adaptability

A Structured process may not be appropriate when it requires adaptive decision-making due to changes in the environment. Adaptation is defined here as adjusting the decision process to accommodate changing environments that cannot be handled through passive isolation (low sensitivity). Structured processes tend to perform well under known conditions, but are not robust outside of these conditions. It is often difficult to produce adaptive behavior using rules because humans do not understand the operational environment, and do not understand how to articulate the rules. Since rules are prescribed prior to operation, based on what designers can *anticipate*, any programmed adaptation tends to be limited by a designer's explicit knowledge.

There has been significant research in the development of "adaptive systems," since adaptation is recognized as an essential function of intelligent behavior [95]. It is often more important for a system to adapt than to optimize [1]. It is helpful to understand what "adaptive" means in the traditional algorithmic context by examining a common application of adaptive decision-making: the control of dynamic systems.

Limits of Classical Adaptive Control

The classical definition of *adaptive control* often implies the continual measurement of the controlled system and its subsequent use in the "self-design" of the controller [150]. (This definition is in contrast to newer classes of automated adaptive processes, which include neural networks). The essential components of a classical adaptive controller are:

1. *Identification* – The measurement of the dynamic transfer characteristics of the controlled system.
2. *Actuation* – The generation of the appropriate actuating signal as the input to the controlled system, based on updated system parameters.

A generic adaptive controller is shown in Figure 2-20. The identifier observes the input-output behavior of the controlled system and provides these to the controller, which updates its internal model in order to generate the appropriate actuating signal (precise excitation is often required). The outer feedback loop is standard to non-adaptive systems.

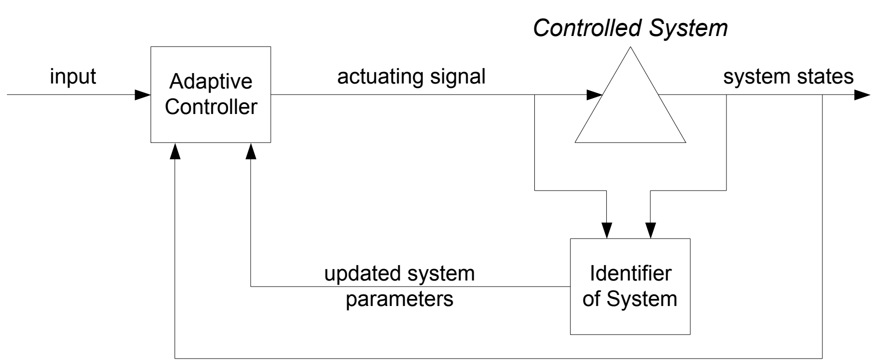


Figure 2-20 Generic model of adaptive control

For controllers of dynamic systems, adaptive controllers are more flexible than their fixed-gain predecessors, but can be an imperfect substitution for the extent of adaptability that is desired. They only adapt to a range of values based on a pre-determined formal model of the controlled system. Classical adaptive controllers merely accommodate *uncertainty* in the controlled system—uncertainty that is limited to a set of prescribed model parameters. In this sense, adaptation is restricted to the self-adjustment of the model parameters, and is also restricted to information that is observed for parameter identification (inputs) and information that is used for control (outputs).

Adapting to Information

While it is difficult to establish general limitations on the adaptability of Structured processes based on their internal processes (e.g., the adjustment of model parameters), it is clear that Structured processes are restricted to the inputs and outputs that are prescribed during design. Hence, Structured processes are inherently limited in the informational sense, by its interaction with the environment.

Since rules operate with prescribed inputs and outputs, *unanticipated* inputs and outputs cannot be part of the decision. Adaptive controllers, for example, are limited by what information can be used for identification, and what information can be used to represent control actions. Any *other* information that arises during operation, as represented by the detached gray arrows in Figure 2-21, cannot be part of the decision [128]. Information may also be missing (perhaps due to a failure) which is illustrated in Figure 2-21 by the detached well-defined feedback. Structured processes are therefore adaptation-limited at least by the inputs and outputs that are prescribed—which prevents the use of additional information, and may cause problems with missing information.

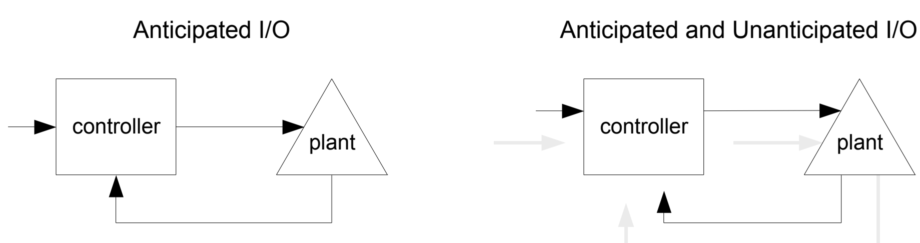


Figure 2-21 Information available during operation

Advantages of Unstructured Processes

Unstructured processes may be more appropriate for adapting to environmental changes primarily because they do not require a decision process to be completely prescribed prior to operation. Both humans and neural networks are often used for their flexibility, and humans are often used in decision systems for their ability to adapt to unanticipated events.

Humans have been long recognized for their ability to adapt to unanticipated situations or cope with contingencies ([43], [67], [165]), and that supporting humans during anomalies is the most important safety issue in the control of physical systems [152]. For example, it is understood that automated controllers have a defined envelope of operation, and humans are often used in a supervisory role in case situations arise that are outside this envelope: such as when equipment fails, or when the environment is no longer appropriate. First, humans can often recognize that adaptation is needed, in part due to their ability to incorporate unanticipated information. In addition, humans are often resourceful, and can devise inventive strategies based on their understanding of a situation. The drive for finding new strategies is particularly critical in safety-related operations, in which human's *survival instinct* provides a natural motivation for using any possible resource for decision-making.

There is debate, however, to the degree in which humans can adapt to largely *unfamiliar* situations. Rasmussen [120] categorizes such "knowledge-based behavior" as different from the more routine modes of system control in which rules are often sufficient. In knowledge-based behavior, "Goals have to be explicitly formulated and plans consciously adopted as hypotheses to be tested." It is believed that humans tend to be successful when the "unfamiliar" situation is, in fact, reasoned to be analogous in some way to a familiar situation . Reason [121] has argued that emergencies which occur in complex systems are often beyond the scope of analogical reasoning, and are therefore largely unfamiliar to the human. In this view, humans may not be valuable for their adaptive abilities.

Neural networks tend to also be much more adaptive than Structured processes, but, like humans, appear to be limited by their experience (training). The modern definition of adaptive systems, in the context of automated information processes, includes the class of self-organizing emergent algorithms [80]. This differs from the classical definition mainly because human designers cannot easily predict the results from their self-organizing rules. Neural networks tend to adapt to information during operation that is a subset of their training (i.e., can tolerate missing information), but cannot incorporate new types of inputs without additional training. This may be critical since training information may not reflect unfamiliar situations. Furthermore, neural networks do not have the breadth of experience that humans have, and may not be able to reason analogically. Hence, neural networks may be superior to Structured processes for adapting to a prescribed set of information (or its subset), but may fall short of human adaptation due to the limited breadth of neural network training.

2.7.7 Miscellaneous External Factors

Pattern Recognition

A Structured process may not be appropriate for recognizing complex patterns of information. Even when the all the relevant sensory data is available, useful information may not be recognized due to the processing required to separate signal from noise, and to otherwise categorize patterns of inputs as belonging to a set of previously encountered situations. This information is often based on visual or aural perceptual patterns, which humans are typically good at recognizing [43]. Simon [141] believes that intuition, which is often described as a mysterious capability of humans, is merely an act of *recognition*. Neural networks also are often good at pattern recognition [83, 90, 147], and many consider this their greatest attribute. However, they do not yet excel at recognizing perceptual patterns such as faces, spoken language, etc. In humans and neural networks, greater experience and training often leads to the ability to recognize larger, more complex patterns.

Contextual Understanding

A Structured process may not be appropriate for interpreting the context of a primary set of information. The understanding of context involves interpreting information differently, based on other secondary information—the circumstances in which information is observed. It is often difficult to understand what this secondary information is, and how it influences the interpretation of information. In fact, Structured processes always are designed based on assumptions of its operational context, and since inputs are fixed, *any* set of rules is inherently limited in this respect.

Humans use contextual information frequently, and rely strongly on it to accommodate the ambiguity of such common activities as conversation. In the context of decision systems, humans are often valuable for monitoring automation to ensure that rules are being operated within its designed envelope—that is, in the correct context.

Category C: Humanistic Requirements

A Structured process may not be appropriate when a decision has functional requirements that seem inherently humanistic. While automation can perform many tasks that were once performed only by humans, there remain functions that appear *uniquely human*—functions that are not usually discussed outside of the context of their allocation to humans. Such decisions often require:

- *Subjective Judgment*
- *Moral Judgment*
- *Creativity*
- *Responsibility*

As with the previous categories (*A* and *B*), the process by which make decisions are poorly understood. This category specifically addresses decision functions that are humanistic.

2.7.8 Subjective Judgment

A Structured process may not be appropriate when the decision requires subjective judgment. Subjectivity is a humanistic quality that contributes to the manner in which different people perceive information, assign preference or value, and incorporate their own Goals to make decisions. Figure 2-22 illustrates these factors as inputs to an Unstructured decision process (some originating internally to the human). Unlike decisions made by humans who are involved in the operational system, *a priori* rules are based on objective descriptions, which, once articulated, assume a reality that is independent of the mind. It may be inappropriate to use rules when decisions involve individual considerations that cannot be represented objectively, such as aesthetic judgments and personal intentions. In these cases, it is important to have humans actively involved in the decision process [1], [22], [99], [126].

Information from human senses may be perceived such that different people have different assessments of the same information. This can be relevant in decision-making for the interpretation of *value*. The personal interpretation of things that are tasted, smelled, touched, heard, and seen often have an associated subjective value (judgment of worth, desirability, significance, importance, usefulness, etc.), which may lead to preferences that are ill-defined and variable among people. Since humans cannot clearly articulate the rules that transform perceived information to value, it may be inappropriate to replace ill-defined subjective judgments with formal rules.

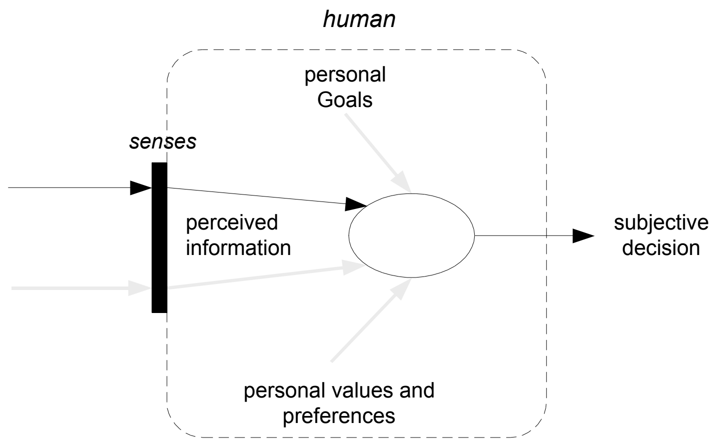


Figure 2-22 Elements of a subjective decision process

Subjectivity can also enter decisions in a "top-down," Goal-oriented manner (capitalization of "Goal" denotes the humanistic definition). A decision system that involves people may be influenced by the personal Goals of the decision-maker, or Goals communicated to the decision-maker by other people. Although the Goals of humans are sometimes well-defined, they are often influenced or even dominated by internally-generated Goals which cannot be represented. Dennet [30] describes these generally as "intentional systems" whose behavior is difficult to explain or predict.

The importance of having humans actively involved in decisions for their subjective value has been recognized as a missing element of classical decision theory [145] and operations research [1], [166]. Ackoff claims that objectivity is not the absence of value judgments, but "…the social product of an open interaction of a wide variety of subjective value judgments." In this view, Structured processes always have a limited ability to reflect the subjective values of those affected by the decision. For example, it may not be appropriate for an algorithm to choose an automobile for a person based on some optimal criteria, without his or her assessments of interior design and comfort, and the exterior color and shape. These "soft" attributes can be important in a decision, and often cannot be articulated *a priori*.

2.7.9 Moral Judgment

A Structured process may not be appropriate when a decision requires moral judgment. Decisions often need some evaluation or understanding of rightness, fairness, ethics, and equity even in the absence their formal definitions. A moral obligation can affect decisions from both personal and societal levels (Figure 2-23). While humans can often provide acceptable solutions to such ill-defined problems, rules are often inappropriate because humans do not understand the complex processes that allow them to make moral judgments in which good and bad are not clearly defined. Hence, humans are valuable for their ability to effectively consider moral issues in their decisions.

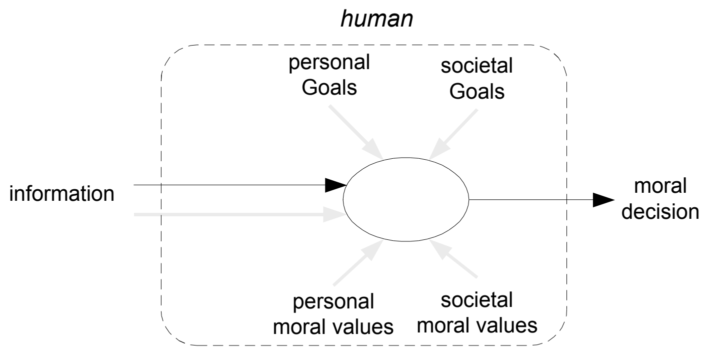


Figure 2-23 Elements of a moral decision process

Moral judgments involve an evaluation of right and wrong, similar to the way subjective judgments may involve a personal evaluation of value. Hence, personal ethical beliefs can be important to factor into a decision. Perhaps more importantly, humans, through social interaction, have a notion of the moral standards for *society*. Decisions at the social level may have to be made by a subset (e.g., managers, politicians) of those who will be affected by a decision. It may be important for the decision to reflect the moral beliefs of *society*, rather than the personal moral beliefs of the decision-maker. For example, juries are expected to discount their personal biases and use moral judgment in a way that captures the spirit of the formal laws of the criminal justice system. Although one can argue that human ethical judgments are arbitrary, rules do not appear to be an effective alternative. In short, the definition of right and wrong can be difficult to explicitly define, but humans are valuable for incorporating moral issues in decisions at both personal and social levels.

Structured processes tend to be limited in their ability to incorporate moral issues. It is often inappropriate to represent moral "cost" in the same way as monetary cost, or other well-defined information. For example, in deciding where to build an airport, one can optimize based on quantifiable information such as material costs and wind speeds, but perhaps not based on whose homes have to be destroyed. Moral issues are important to consider, but tend to be particularly incommensurate with other information.

2.7.10 Creativity

A Structured process may be inappropriate when the decision requires creativity. This can be desired for various purposes: design, discovery, invention, art, music, gaming, etc. Rules, which tend to be most valuable for rigid, repetitive decisions, are often ill suited for generating *novel* decisions. Being original nearly defines being relatively unconstrained, suggesting that the inherent constraints of rules place limitations on the outputs of a decision process. One manner in which Structured processes are constrained is through the representation of the decision *output*, illustrated by the gray *output* arrow in Figure 2-24. However, even when the decision space can be represented *a priori*, humans do not understand the complex processes that are involved with creative decision-making.

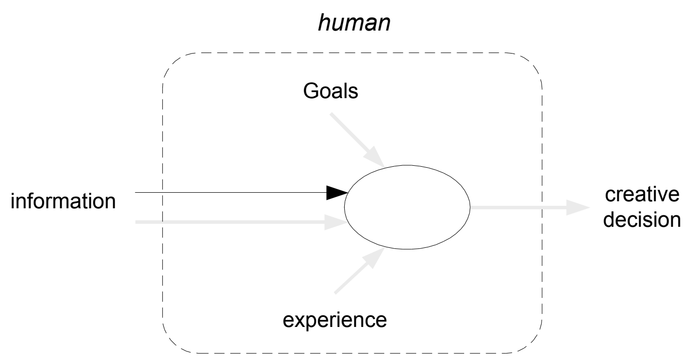


Figure 2-24 Elements of a creative decision process

Two Views of Creativity

Many believe that creativity is the essential difference between mind and machine; computers only do what they are programmed to do [19]. In this view, humans can be original, inventive, and imaginative; they have the natural ingredients for the spark that ignites their creative powers.

Another view is that creativity is not a special attribute of humans. Creativity is a matter of effectively searching through a very large decision space— a task that Structured processes can perform. However, this latter situation seems to require a "well-defined *problem*."

Creativity as a Search Process

Rules can offer creative value to decision-making when the problem is at least "well-defined." That is, when a definite criterion exists for testing if a proposed solution is acceptable

[97]. Furthermore, when a problem space exists in which the initial state, goal state, and transition states can be represented, decision-making is simply a matter of searching through this space. The difficulty, of course, lies in the size of the search space [141], [158].

In this section, creativity is discussed as a paradigm for searching among a large set of well-defined decision options that can be formally tested with rules. Some examples of creative decision-making in this paradigm are:

- A novel chess move
- A proof of a mathematical theorem
- A non-intuitive combinatorial design
- The discovery of new relationships from observed data

These creative solutions have been found using Structured processes. While searching may be performed by computers at high speed, random search is often impractical (about 10^{120} options for chess), so that the key issue is to search *efficiently*.

Consider the combinatorial mechanical design of a legged robot from a set of pre-defined joint modules. Given the appropriate constraints (e.g., kinematic, power), and a method for formal evaluation (energy consumption, mobility), rules can be used to try different configurations and evaluate them. For efficiency, evolutionary ("genetic") algorithms, may be used to weed out unpromising options in early "mutations." After many "generations," superior designs tend to emerge (although optimal designs cannot be guaranteed). Humans tend to be good at quickly finding acceptable options, but may overlook configurations that are non-intuitive. This example illustrates that creativity is not necessarily restricted to humans.

Advantages of Unstructured Processes

It appears that Unstructured processes are most valuable for creative decisions when the goals and allowable "moves" cannot be explicitly defined. This includes situations when it is not possible to formally test decision alternatives, as in optimization. However, even when the goals are well-defined, the means to achieve these goals—that is, the decision outputs—may not be well-defined. In such cases, the "search" paradigm is not appropriate for considering creative decision-making.

Ambiguous goals include those that require aesthetic evaluation (music, poetry, fine arts, fashion, industrial design), moral evaluation, or any situation which does not have a well-defined objective. Being creative is not just generating wild options, but generating novel yet *reasonable* options. Humans are often needed to evaluate whether a decision is reasonable.

A separate issue from the ability to test a decision alternative is the ability to represent the decision alternative space. If this space is definable, it is a search constraint in itself. In determining the robot configurations, for example, the possible options are bounded by the representation of the problem, such as the design parameters. While decisions typically have some degree of constraints on the outputs, it may be possible to consider other alternatives outside of any particular representation. Since an Unstructured process is not constrained by a representation of its output, it is not inherently limited in this respect.

Humans are often recognized for their creative abilities. It is difficult to imagine computers writing compelling stories, composing fugues, designing buildings, inventing new products, or discovering a cure for cancer—at least not without a human involved. It is apparent that the underlying thought process which provides such enlightenment remains elusive, suggesting that creativity is the result of a process that is poorly understood and not amenable to rules [79], [100]. Rather, it is suggested that computers are useful tools for exploiting human creativity by means of human-machine symbiosis [24], [45], [132].

2.7.11 Responsibility

A Structured process may not be appropriate when there are social demands for responsibility in the decision. When an individual is responsible for a task, he or she given decision freedom in exchange for an assumed reliability in carrying out that task, and are held accountable or answerable for the decision. Some believe that responsibility should be the primary factor governing function allocation [85]. Jordan [67] believed that responsibility is humanistic, and stated that "…we can never assign (machines) any responsibility for getting the task done; responsibility can be assigned to (humans) only." Here, responsibility is examined not only in the context of its human or machine embodiment, but also for its effects on a decision process independent of its allocation.

While the importance of responsibility in decision-making is known, the *process* by which humans incorporate responsibility into decisions is not understood. Minksy distinguishes "local" and "global" responsibility, the latter being the more difficult to understand [96]. Responsibility is one way of propagating the values and goals of other people—those not actively involved—into the decision process. If the decision-maker is held accountable for decisions, there is a social belief that the decision will be reliably carried through, will not be negligent or selfishly made, and will consider the goals of other people. This belief is based in part on an assumption that the decision-maker understands that future decision consequences will, through societal feedback, reward or punish him or her accordingly. With this knowledge, illustrated as inputs in Figure 2-25, a decision-maker is believed to act responsibly by making decisions that reflect the interest of others. In addition, responsibility may be required for placing blame or reward after the decision consequences are known.

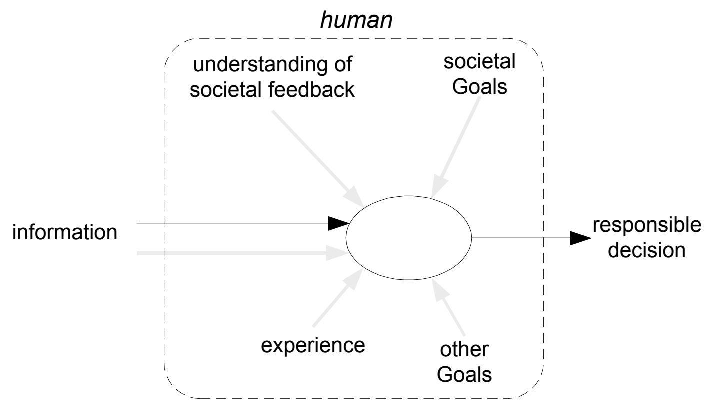


Figure 2-25 Elements of a responsible decision process

Consider a few examples of how responsibility might affect a decision (moral issues aside). A physician may take extra care in prescribing medication because of an understanding of a malpractice lawsuit. A car mechanic will be sure that the brakes function properly because of the risk of job loss. A politician will attempt to make appropriate decisions in order to be re-elected. It appears that when people are held responsible for their actions, there is often an associated collective benefit to certain groups. The concept of responsibility is therefore important to society because it allows decision-makers to act independently—without monitoring or supervision—and at the same time creates a mechanism for improving the quality of decisions.

Goals and the Survival Instinct

An important ingredient of responsible decision-making is an understanding of other people's goals. While rules can be designed to accommodate goals, there is comfort in knowing that a human decision-maker understands other people's goals, and that these goals will be considered no matter what situation arises. Hence, if a situation arises that is beyond the assumptions of the rules, it is desirable that a human will not blindly execute these, but will reformulate a new set of decisions based on his or her understanding of the goals and values of others.

As mentioned earlier, an important common goal is people's *survival instinct*. This instinct motivates people to use whatever resources possible, certainly beyond any prescribed rules, in order to ensure survival. This is perhaps why people feel comfortable knowing a human is "in charge" of decisions [138], such as piloting aircraft or performing surgery—particularly when the decision-maker's life is at stake.

Part of responsibility is maintaining an understanding of what other people want—an important goal being the desire to survive—and incorporating this knowledge into decisions in a constructive manner. This can be difficult to incorporate into Structured processes.

Tracing Responsibility

In addition to the sense of improved decision-making, responsibility is often required due to *a posteriori* issues—after decision consequences are known. People are often required to be legally or morally responsible, to accept blame or reward, or to provide explanations in the event of an incident such as a catastrophic error. A Structured process—whether allocated to human or machine—may not be a sufficient process because responsibility can be further traced to those who articulated the rules (e.g., designers, trainers) or those in charge of their development (e.g., managers). When decisions are through a Structured process, even a human decision-maker can then claim that he or she was just following standard procedures—allowing them to "pass the buck." Therefore, rules may not be appropriate when decisions require responsibility that cannot be further traced.

2.7.12 Miscellaneous Humanistic Requirements

Understanding Goals

An important attribute of human decision-makers is their ability to understand the Goals of others. This is one part of their complex mental model of human behavior. It is particularly important when unanticipated changes occur during operation, such that the assumptions governing the rules are no longer valid. In such cases, it is valuable to override rules, formulate new Goals, and continually determine what parts of a situation are most important. The understanding of Goals is also fundamental to moral and responsible decision-making, but is generally important for all decisions in which the goals are not well-defined, but the decision affects other people.

Understanding Intent

The understanding of intent is similar to goals, but here refers to a generalization from observed actions of a decision-maker. An example generalization is the estimate of intended information desired from a keyword search. Humans can often easily understand what is required from these keywords, although the mental processes that allow this are not well-understood. Understanding intent allows people to generate "top down" knowledge of other people's Goals based on what is often a sparse set of observations, which can be valuable when tasks involve communication with other people.

Category D: Implementation Issues

A Structured process may not be appropriate when considering the issues for its implementation. A process that is theoretically sufficient may not be practical due to the physical resources available for its operation and support. Implementation issues include:

- *Information Cost*
- *Processing Resources*
- *Errors and Robustness*
- *Design, Verification, and Maintenance*

In the previous categories, operational decision-making issues were addressed by considering processes in an abstract sense. Here, broader systems issues associated with implementing a process are considered. The implementation issues discussed here have relevance to the allocation of Structured processes between humans and machines, but do not cover *general* human-automation issues in which a Structured/Unstructured distinction is not relevant.

2.7.13 Information Cost

A Structured process may not be practical to implement due to the cost of obtaining information. "Cost" is discussed in the context of rules, whether allocated to humans or automation, and can imply allocating resources beyond direct economic costs, such as time, effort, reliability, safety, etc. Since a Structured process requires a known set of information, the cost and benefit of obtaining this information can be explicitly considered when determining if a Structured process is appropriate.

Information may be costly to obtain for many reasons. Classical decision theory relies *on a priori* utilities, but these can be costly to elicit. Similarly, knowledge bases in expert systems can be costly to elicit from experts. Other considerations include the cost of accessing databases, and hardware issues such as sensors and communications networks. In addition, Structured processes may also require that information is processed or placed in a specific format, which can require significant effort.

Advantages of Unstructured Processes

Although Unstructured processes also require certain information, this can be different than the information required by Structured processes. Adequate decisions can often be made when information varies or is missing. While Unstructured processes may have an associated set of information that makes the decision "informed," they do not necessarily *require* a predefined set, and may rely on other, less-costly resources. For example, a physician may be able to informally screen for a skin disease through visual inspection, where a formal diagnosis would require an expensive biopsy.

2.7.14 Processing Resources

Computer Processing Limits

Due to finite processing resources, it is important to understand that not all Structured processes may be practical to realize. The solution to a large set of equations, or the exhaustive search through a large solution space can require impracticably large memory, clock speeds, in order to arrive at a decision in a reasonable time. Hence, even when a Structured process is theoretically sufficient, it may not be practical due to processing resource limits.

To illustrate the criticality of these resource issues on algorithms, consider the growth of digital signal processing (DSP) in the last few decades. Before the 1960's, frequency domain analysis was performed on computers using the discrete Fourier transform (DFT)—a robust, but inefficient algorithm. For the resources at the time, DFT applications were limited to the postprocessing of data. The invention of the fast Fourier Transform (FFT) allowed the same results to be calculated with much less effort (for a data array of length n, the number of calculations required dropped roughly from n^2 to $n \log(n)$ assuming only minor constraints on the array size). Suddenly, computer resources were not nearly as limited, allowing DSP to become real-time, and eventually mainstream technology (e.g., consumer products). This example illustrates that the physical realization of a Structured process can be strongly dependent on the computing resources that are available.

Cognitive Processing Limits

Humans also are limited in their ability to process information. While it may not be fitting to compare cognitive resources to digital computers, research in human information processing has demonstrated the existence of approximate limits in certain types of problem solving. Assuming an information-processing model of cognition that is composed of parallel and serial parts (as in [120]), it appears that these limits apply primarily to the serial parts.

Simon [140] provides a review of the human limits revealed by certain serial processing tasks. Of particular significance is that attention and short-term memory are limited to about seven "chunks" (familiar items, like numbers or words) of information [93]. These observations are important because the execution of rules often requires deliberative thought that may be limited by attention and short-term memory. For the execution of well-defined rules, computers are not nearly as limited.

Advantages of Unstructured Processes

When a Structured process requires a prohibitive amount of processing, an Unstructured process may be able to perform the same *function* with available resources. For example, an exhaustive search for an optimal solution may be replaced by a strategy that finds an acceptable solution with less work, based on ill-defined heuristics.

2.7.15 Errors and Robustness

While it is recognized that Structured processes tend to be less robust than Unstructured processes—without considering their allocation—the implementation of these processes has a further effect on robustness. Specifically, Structured processes are often sensitive, such that noise and errors associated with the inputs and rules can propagate to large errors at the output [81],

[128]. Therefore, noise and errors that are the result of implementation can be important to consider. For example, computers have associated quantization errors, bit errors, and electromagnetic noise, while humans are affected by boredom, fatigue, and emotions.

Advantages of Unstructured Processes

The robustness of Unstructured processes is also influenced by the processing machinery. For instance, in connectionist systems such as the human brain or artificial neural networks, the failure of single neuron out of a thousand will not likely affect the output. In contrast, traditional computer hardware and software is typically optimized for serial operations, such that a single bit error in a Structured process can easily propagate, and never be absorbed. In such cases, fault tolerance is often achieved through parallel implementation.

2.7.16 Design, Verification, and Maintenance

A Structured process may not be appropriate when considering the necessary support for its design, verification, and maintenance. These are broad issues that extend beyond the period of operation.

Design and Development

For complex tasks, a Structured process may require considerable effort in design. It should be no surprise that software development is a significant cost of the overall development of decision systems. Similarly, human decision-makers may require training to learn rules. The development of complex decision systems further requires that sub-processes that were designed individually must also work collectively.

In contrast, Unstructured decision processes do not require explicit rules, and might be developed with less effort. For example, humans can develop their own internal rules with practice, or may naturally possess the relevant skills (e.g., image recognition). Neural networks can often be easily trained if the appropriate data is available.

Verification

It is often important to consider the verification of a Structured decision process for operational use. Verification is generally a formal evaluation of a decision process in which a decision process is deemed acceptable for future operation. It can be particularly important—and is often required—in safety-critical systems, such as medical and aircraft decisions.

For reasonably complex traditional software, verification typically requires simulation and/or prototype testing to generate information for evaluation. Both can be costly, but are usually necessary. However, it may not be possible to verify *highly complex* designs with a reasonable amount of simulation or testing. Hence, advanced algorithms such as those developed in artificial intelligence may not be used due to verification difficulties, even when designers intuitively have high confidence. This problem has been discussed extensively in the context of expert systems [51], [169].

Based on the same logic, it would seem that Unstructured processes would also be difficult to verify. However, humans are often used in decision-making with much less formal verification. For instance, they may only require certification in a subset of conditions that automation would require. There appears to be confidence in human decision-makers, even though they may not generalize correctly during operation. The same level of confidence does not generally hold for neural networks.

Maintenance

It is often important to consider the maintenance of a Structured process, particularly when changes are periodically required. While a simple set of rules can easily be changed with known implications (e.g., adjusting a threshold or gain), seemingly simple changes to complex processes can lead to unexpected behavior, especially as modifications are accumulated. Hence, re-verification is often required. In expert systems, process upgrades may require additional elicitation of rules from experts, and it is very difficult to understand the implication of changing rules in a large data base [90], [169]. When Structured processes are implemented by humans, new procedures have to be learned, which may also require re-verification. Furthermore, rules and procedures in decision systems often are supported by an extensive amount of documentation (e.g., operating manuals) which has to be updated when modifications occur.

Advantages of Unstructured Processes

Unstructured processes may have some advantages for maintenance reasons. Humans tend to adapt to minor changes without retraining, and can understand changes in *functional* requirements without needing to understand how the decision process will be affected. For neural networks, it may be easier to retrain with new data than to understand and modify code in a traditional algorithm—particularly when the people responsible for code modification are different than the original designers. It may be easier for humans to select the data for training, than to explicitly update rules.

2.8 CONCLUSIONS

This chapter introduced the concept of semi-Structured processes. The semi-Structured framework, which consists of definitions, diagrammatic notation, and organizing principles, is intended as a tool for analysis. It provides a way to view decision systems as information processes in which both well-defined and ill-defined components can be explicitly considered.

An important issue discussed early in this chapter is that a process needs to be appropriately matched to the environment in which it operates. While simply stated, this is perhaps the greatest challenge to engineering design—particularly the design of *decision systems*. The reason for this is primarily because people do not explicitly understand, during the time period of design, how to make decisions at a future time: during "operation."

Structured processes provide a means for exploiting what *is* understood prior to operation. These processes can be realized by humans, such as with standard operating procedures, but are often valuable because they can be reliably automated. In the context of decision system design, Structure represents the part of the operational process that is constrained by the design choices. While Structure is often desirable because it allows system designers to completely prescribe how a decision is to be made, rules are always limited based on what is understood at design. Section 2.7 provides a comprehensive list of these limitations.

Given that decision systems are always designed with some degree of uncertainty or lack of knowledge, it may be desirable to incorporate Unstructured decision process into a system design. By doing so, it may be possible to account for what is not understood prior to operation by using decision processes that are determined during operation. It is believed that humans add value to Unstructured processes, such as with judgement and intuition. Automated algorithms such as neural networks also can be considered Unstructured, but these are limited in their "experience" (training data) and their ability to take into account humanistic requirements such as subjective judgment.

In decision system design, there are many ways to use humans and automation. In order for a decision system to be matched to its operational environment, such that decisions are appropriate, it may be necessary to incorporate Unstructured processes in the design. This may move the system away from optimality during nominal conditions, and add a degree of uncertainty in *how* a decision will be made, but it may also provide robustness in functionality by allowing the process adapt to meet the goals of the decision system. It is ultimately up to the judgment of the designers to determine the extent to which a process is Structured; the extent to which a system is determined prior to operation.

The semi-Structured framework provides a way to understand the implications of design choices in part by allowing design concepts to be explicitly considered in terms of their Structured and Unstructured components. In particular, the representation of Unstructured processes may help prevent designers from overlooking the ill-defined but important aspects of decision-making. Furthermore, Structure can then be designed in part to support Unstructured decisions. This insight may improve the way humans and automation are used within a decision system.

CHAPTER THREE

ANALYSIS OF EXAMPLE DECISION SYSTEMS

3.1 INTRODUCTION

This chapter applies the theoretical concepts of the previous chapter to example decision systems. The primary purpose of this chapter is to provide additional insight by analyzing existing designs within the semi-Structured framework.

The decision systems chosen for analysis here are highly evolved designs. That is, while their decision process cannot be formally proven to be *a priori* optimal (due, in part, to the presence of the Unstructured sub-process), the systems here are assumed to have emerged over time—based on "survival of the fittest" reasoning—because they are good. With this assumption it is not necessary to justify *that* a design is good, but it may be useful to understand *why* it has evolved in a certain way.

The choice of examples are primarily based on three issues. First, they are chosen to be representative of a larger class of systems, so that results can be generalized to other systems. Second, simple systems were desired because they capture some of the important characteristics of more complex systems. Third, examples were also chosen based on their semi-Structured "topology": the order of sub-processes in the primary information path.

3.1.1 Semi-Structured Process Topologies

Figure 3-1 illustrates the four basic topologies used in this chapter for categorizing decision systems, based on the diagrammatic notation introduced in Chapter Two. Each topology is characterized by the sub-process order, such as "*Unstructured into Structured*" or "*U-S*." Topologies not only serve as a reasonable basis for classifying decision processes, but they potentially add another dimension of insight by showing "where" Structure resides in a system.

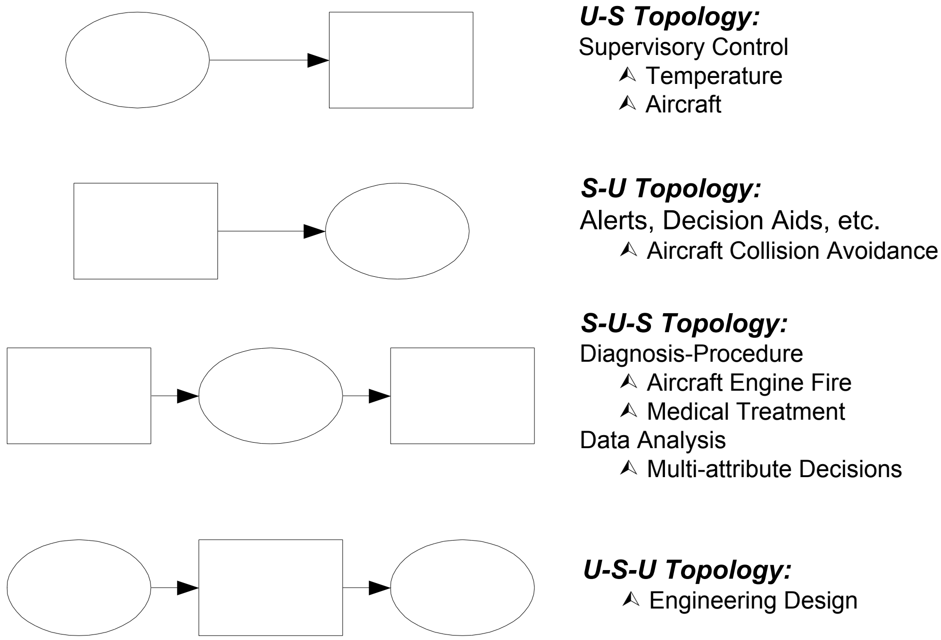


Figure 3-1 Overview of decision process topologies

The topologies in Figure 3-1 reflect a variety of decision systems. Since these represent processes, the diagrams do not contain information about *functions*. However, the location of a Structured process often indicates its function in a general sense.

When Structure is on the left of the process, or near the *beginning* of the primary information path, it is often to act as an "observer" by providing information. When Structure is on the right, or near the *end* of the information path, it is often operating as a "controller." Even when the controlled system is informational vs. physical, these concepts may still apply. In human-automation systems, automated rules are valuable for both informing humans about the state of the controlled system, as well as translating commands to perform low-level control. Structure is used primarily in *support* of Unstructured processes.

3.2 TEMPERATURE CONTROL

The temperature control of a closed system is an example of a semi-Structured decision process. This example is chosen for analysis because it is a *highly evolved* example of a "supervisory control" system, which can be modeled as an *Unstructured-Structured* topology. This is shown in Figure 3-2 without feedback to illustrate the primary information path. Supervisory control systems are a common type of decision system in which humans intermittently program and continually receive information from a computer that itself closes an autonomous low-level control loop through artificial sensors and actuators [138]. Temperature control is also chosen because it is simple, yet it can illustrate some of the attributes of more complex systems.

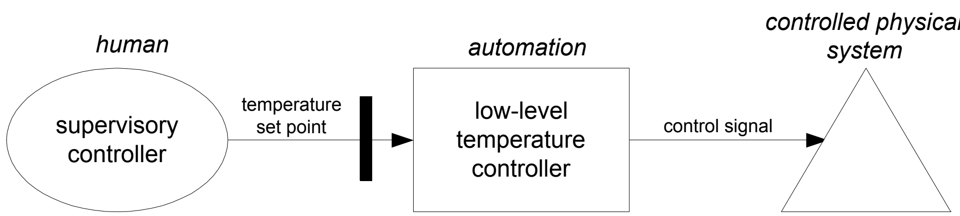


Figure 3-2 Simple model of temperature supervisory control system

Figure 3-3 illustrates the same system at a higher level of detail. The set point represents a well-defined state—temperature—which automation controls using well-defined rules. Both human and automation use temperature feedback for making control adjustments, but the human also uses feedback and other information, some of which is not well-defined. The function of the human supervisory control process is often ambiguous. In contrast, automation has a well-defined function: to control temperature to the value represented by the set point. This diagram shows the variety of inputs and outputs within the system, as well as the important interfaces. The following sections refer primarily to this diagram.

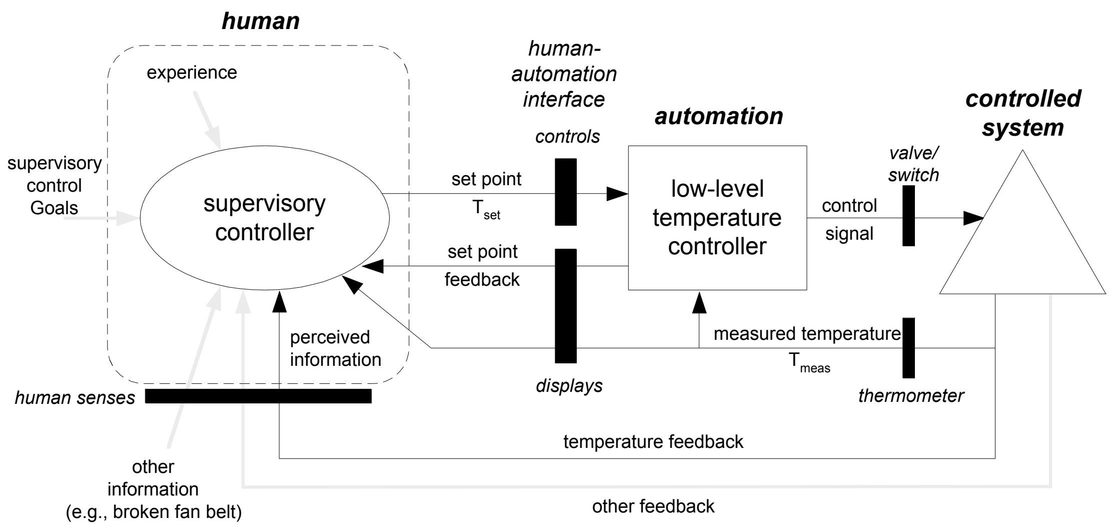


Figure 3-3 Detailed model of temperature supervisory control system

3.2.1 The Low-Level Temperature Controller

Automation is believed to be used primarily for off-loading the cognitive and physical work associated with low-level temperature control. Automation adds value in part because it allows humans to interact less frequently and more naturally.

Low-level control is an appropriate use for automation in a temperature control system, primarily because the process can be reduced to a well-defined set of rules. In this simple, single state example, the rules for controlling temperature about a set point are well established. An example set of temperature control rules are the following:

IF [$(T_{set} - T_{meas}) < 0$]

THEN [*activate heater*]

ELSE [*deactivate heater*]

where T_{set} = set point temperature, and T_{meas} = measured temperature from a thermometer. A differential signal or temperature error is calculated from the two inputs, which produces an *on* or *off* action based only on the sign of the error. The above algorithm, illustrated in Figure 3-4, is very simple, and is used here only to illustrate that *low-level* temperature control can be satisfied with well-defined rules.

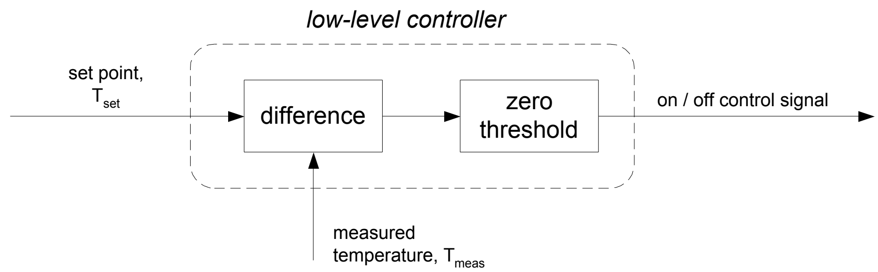


Figure 3-4 Decision logic of a simple low-level temperature controller

In order to understand why rules are appropriate for low-level temperature control, consider the following attributes of this part of the decision system:

- *The goal of low-level control is well-defined* – The goal is to control the measured temperature T_{meas} to a value represented by the set point T_{set}.[4]
- *The inputs are well-defined* – Both the set point and measured temperature can be unambiguously represented.
- *The outputs are well-defined* – The control signal represents an unambiguous actuator state.

The above properties are consistent with the discussion in section 2.7. Although the semi-Structured process collectively may have to deal with more complex issues beyond the capabilities of the low-level controller, such issues can be accommodated by humans. Supervisory temperature control processes are decomposed such that a human provides a well-defined temperature set point, T_{set}, which automation then uses as one of two inputs. The design of low-level feedback control processes typically assumes the target state parameter is given. Here, T_{set} is explicitly recognized as the output of an Unstructured decision process.

3.2.2 The Supervisory Controller

The human supervisory control function, shown isolated in Figure 3-5, is examined as an Unstructured decision process. Given the constraint of a required set point by automation, the human operator must use this parameter to satisfy the system *goals*—beyond the well-defined function of automation. To satisfy these goals, the human needs observe a more complex set of *information* than the automation observes, and also have the capability to satisfy ill-defined

[4] Systems often use hysteresis to control temperature within a small region about a set point.

functional requirements—such as considering subjective issues and adapting to unanticipated conditions. The combination of these functional requirements make the supervisory control function inappropriate for rules, suggesting that humans are valuable parts of the temperature control decision process.

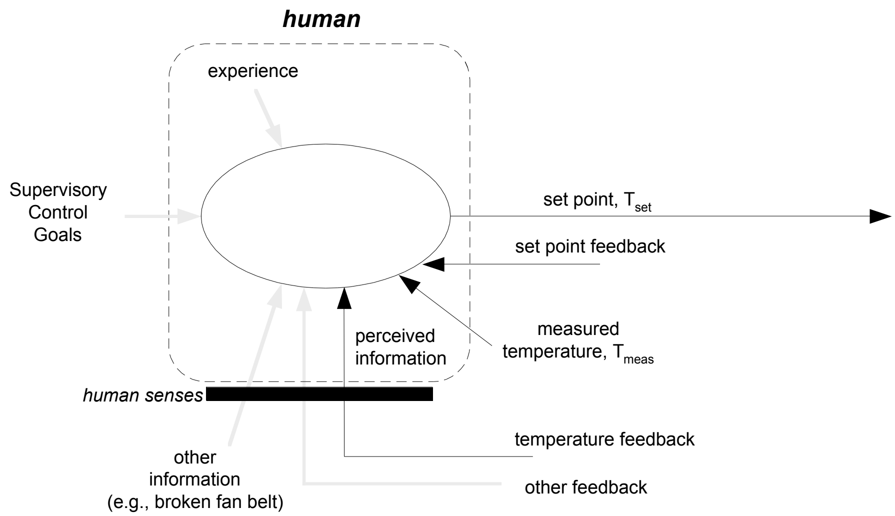


Figure 3-5 The Unstructured portion of temperature control (from Figure 3-3)

Goals

The human operator is valuable because the goals of the system often cannot be explicitly represented. Although temperature is a well-defined state, there are a number of other states affected by the set point decision—some of them ill-defined individually, or when considered jointly (e.g., trade-offs). The goal of the decision system is more than controlling temperature to a specified target; Goals may include many other states that are altered by a temperature decision.

The goals of a system can be ill-defined in part because of the need to consider subjective issues. When the temperature-controlled system affects people—as in building heating systems—a priority is often placed on comfort, which is an ill-defined, dynamic function of temperature and perhaps other parameters. Such subjective issues are not considered by the low-level controller.

Supervisory control Goals also involve other system states that are the result of a set point decision. One such example is the definition of trade-offs between comfort and its associated cost. People may find it difficult to explicitly articulate how much money it is worth to increase the temperature a few degrees, since these attributes are incommensurate. However, humans can easily consider these issues to some extent. The ability to consider Goals without their explicit representation can make humans a valuable part of the operational decision process.

Information

Humans are valuable in the temperature control decision system because of the rich set of information that they can access. Whereas automation may require well-defined information, such as from a themometer, humans sense and interpret meaningful information that cannot be explicitly represented. This information, illustrated in Figure 3-5 as multiple inputs to the Unstructured process, allows humans to make a more informed decision, particularly with respect to the ill-defined system goals.

Consider temperature feedback from the controlled process. As Figure 3-5 shows, feedback is used for both the supervisory control decisions as well as low-level control. Temperature feedback is not merely redundant. While the measured temperature is available to both processes, temperature is also directly *perceived* through the human senses, and is valued subjectively with respect to the Goals. Perception of temperature is often critical since a display of temperature through the human-automation interface may not be as meaningful for a given task, such as those involving subjective judgments.

Humans also observe ill-defined feedback states that are affected by temperature. These states, while not controllable *directly*, may also be important to consider in the temperature set point decision. For example, perceptual processes may be required to recognize relevant states that are affected by temperature: the sound of a steam engine, the viscosity of a resin, or the browning of a crust. Such information often has perceptual value beyond what can be observed through formal processes.

Lastly, an important function of automation is to provide additional information to the human supervisory controller. This information is designed into the system by means of a human-automation interface.

Adapting

A functional requirement of a supervisory controller is often to *adapt* to unanticipated situations. The dynamics of any situation can result in unanticipated changes that are difficult to accommodate with rules: goals, information, and changes to automation and the controlled system. It is common for humans to successfully adapt to unanticipated conditions, since the decision process is not determined until operation.

Humans appear to be valuable because they can easily adapt to available information. Consider when an input, such as perceived temperature, is missing. While this may make the decision under-informed, people can rely on other sources of information and experience in order to make an adequate decision. For example, humans detect a broken fan belt perhaps by adapting to temperature and noise patterns. Similarly, *additional* information that is operation-specific can also be incorporated by humans. In contrast, automation is limited to only the information for which it is designed.

Humans also adapt to changes in the environment. Consider what happens when a thermostat drifts out of calibration. Humans do not necessarily recalibrate, but adapt by compensating with a set point adjustment. This is possible in part because humans perceive temperature directly, without relying completely on the Structured process of the thermometer (which changes over time). Whereas the rules of the automated process operate on sensor information independent of its representation, humans can easily alter their decision process to accommodate external changes. Furthermore, experience and knowledge can be used in understanding *how* to adapt.

3.2.3 The Human-Automation Interface

The purpose of this section is to understand the role of the interface in the semi-Structured temperature controller, particularly with respect to the human's Unstructured process. The interface is necessary for providing the temperature set point, but it is also important for providing the proper form and content of information to the human. The human-automation interface shown in Figure 3-6 is a common design for the temperature control of buildings, and will be the baseline example for this section.

Representing Information

The design of the human-automation interface has evolved to accommodate the communication of three parameters—the current set point (T_{set}), feedback of the set point, and feedback of measured temperature (T_{meas}). It provides a clear representation of the three parameters, allowing humans to interact intuitively.

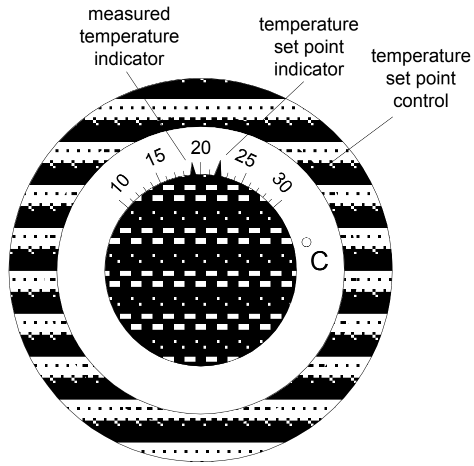


Figure 3-6 Example human-automation interface for temperature control of buildings

The interface in Figure 3-6 provides a means for adjusting the temperature set point by rotating the transparent dial. The position of this dial—the set point value—is displayed by a dial indicator which is referenced against a fixed, calibrated scale. In addition, measured temperature is indicated on the same scale with a different arrow. These two feedback indicators provide both an absolute and relative frame of reference. While this information is not necessarily *required*, it is assumed from this highly evolved interface design that set point feedback and measured temperature feedback add value to the Unstructured decision process by making the decision more informed.

Set Point Feedback

Feedback of the set point is important for a number of reasons. First, it provides humans with a reference for understanding the state of automation—an indication of its future behavior. Without this reference, it is more difficult to understand how control inputs will affect the temperature. Furthermore, set point feedback allows people to use experience (also illustrated as an input in Figure 3-3) more effectively by memorizing past settings. Differential input devices, such as unmarked dials, do not provide this information.

Although the *form* of set point feedback varies among systems, interfaces often provide a unique representation of the temperature set point. For example, the representation of colors (e.g., blue to red), or convenient numerical scales (e.g., 1 to 10) may be appropriate when an absolute reference is not required. Otherwise, feedback in terms of a standard calibrated scale is used, which is the case in Figure 3-6. In either case, it appears valuable to provide the operator with a unique set point representation.

Measured Temperature Feedback

The display of measured temperature also appears valuable in the set point decision. The temperature display can be important when monitoring automation, especially when the operator is physically removed from the closed system. In fact, even when the operator has the ability to sense temperature naturally, this observation can be inaccurate due to the subjectivity of their direct perception.

Measured temperature feedback is often represented similarly to set point feedback. Since both parameters represent temperature, identical forms allow for a more natural comparison. This can be beneficial when making adjustments *relative* to current conditions (e.g., "hotter"), as in differential temperature control.

Closing Remarks

Figure 3-7 summarizes the well-defined information that the interface is designed to accommodate. Interfaces such as those in provide humans with two parameters, T_{set} and T_{meas} , to support their Unstructured decision process. In this case, automation is simple enough that the set point feedback provides humans with a sufficient understanding of future behavior. This may not be the case in more complex temperature control systems.

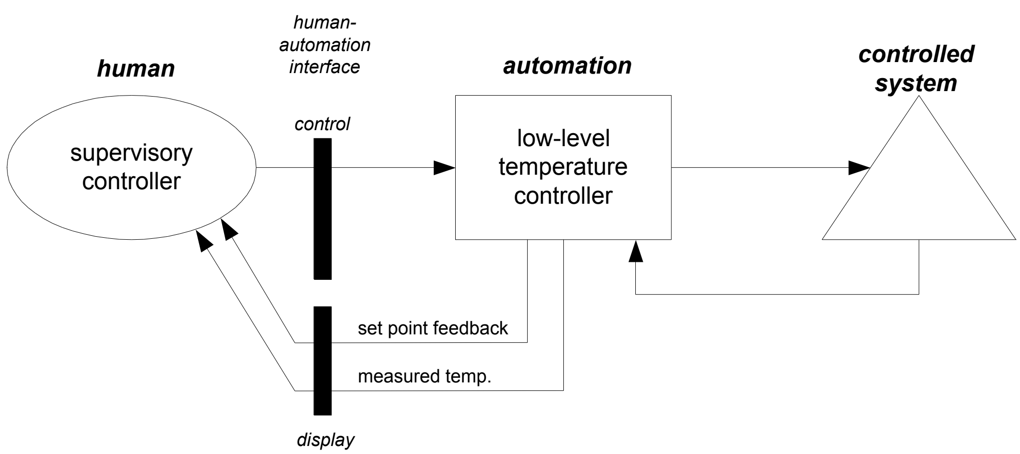


Figure 3-7 The interface is designed to support human supervisory decisions

3.2.4 Implications of Additional Structure

In some cases, it is appropriate to automate a larger portion of the decision process. In these situations, the human—if used at all—is at an even higher level of supervisory control. More automation means that more of the decision process is determined prior to operation, which generally requires additional assumptions about the operational environment. This, in turn, is reflected by a more complex interface.

Programming a Temperature Profile

A common way to incorporate more automation in the supervisory temperature control system is to have automation adjust the set point based on well-defined criteria. This criteria is often the time of day, in which case humans program a temperature *profile*. A prescribed temperature profile may not precisely reflect the way humans adjust temperature, but under certain conditions is a reasonable way to automate over longer intervals.

Figure 3-8 illustrates a possible decision system in which the temperature profile can be programmed for automation (for simplicity, portions of the system are not shown). For the temperature control of buildings, a profile might be to decrease the temperature at night. The automatic mode, which can often be switched off (to the baseline "manual" mode), is considered a three-level functional hierarchy in which the bottom two levels are automated. In fact, the lowest level can remain unchanged from the baseline case discussed earlier. However, the human's Unstructured decision process as a supervisory controller changes to accommodate the higher level of automation.

Note that even a small increase in automation can substantially change the complexity of automation and the human-automation interface. Humans are required to choose the mode of operation, which provides a new opportunity for errors [5]. In addition, "auto" mode requires profile information, which is more difficult to communicate than a single set point. Lastly, the complexity of the additional automation is reflected in the feedback of its internal states. Mode, time, and profile are among the information that people need to make an informed decision, resulting in a more complex interface.

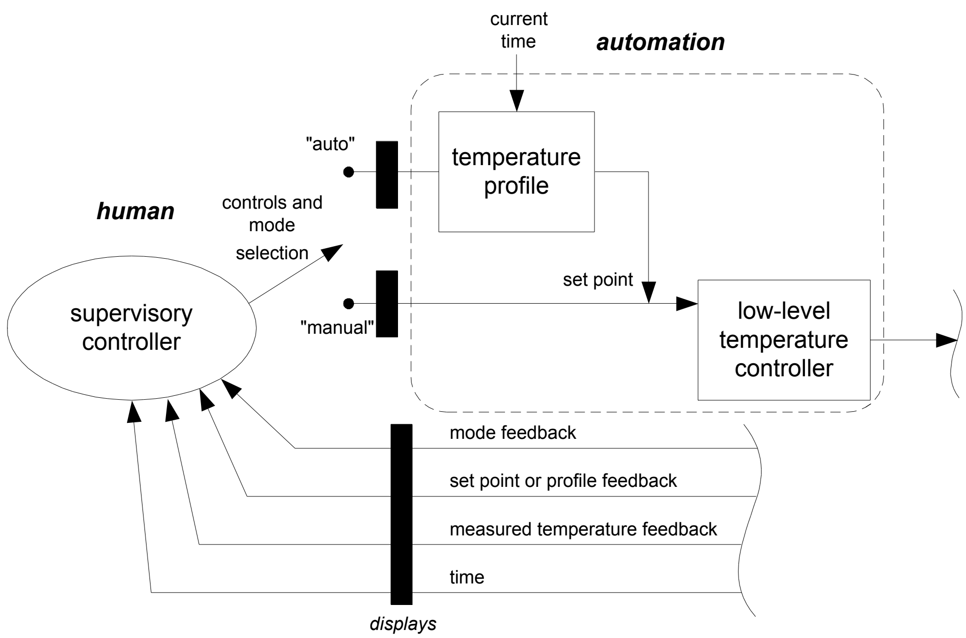


Figure 3-8 Highly automated temperature controller

The assumptions that allow temperature profiles to be automated are more constraining. During these automated periods, it is not possible to incorporate new information, subjective judgments, or modified goals without human intervention. Furthermore, since humans interact less frequently, they may have less understanding about the behavior of automation—a problem that is amplified by the additional complexity.

Full Automation: Automobile Temperature Control

"Fully automated" temperature controllers require even more assumptions in order for rules to be appropriate. As an example, automobile engine temperature controllers are a Structured process: when the coolant temperature exceeds a threshold, a fan draws cool air through a radiator. Humans are essentially not functional in the operational decision system. Assumptions for automated control include stable thermal properties (e.g., coolant properties, water pump flow rate, radiator performance) and a prescribed range of environmental conditions (e.g., ambient air temperature, altitude, towing load). Automated temperature control need not deal with ambiguity or require humanistic considerations. In short, a fully Structured process is appropriate here because temperature control in these environments are well-understood.

In a broader view, automobile engine temperature controllers use humans in the decision loop. Most cars provide a display of engine temperature, or at least an overheat warning light to inform the operator of anomalous conditions. Human operators may not be able to alter the controller's decision logic, but are able to modify operation by driving slower, turning off the engine, or filling the radiator reservoir. When the assumptions of the rules are overrun—perhaps due to neglected maintenance or failed components—human actions can prevent dangerous or costly consequences from excessive overheating. While such actions can be rule based, they may be costly to implement (a critical issue with automobile), and may require judgment for their proper use. For example, it can be dangerous to automatically shut down a car in the middle of traffic; humans can understand the broader issues and risks, and can shut down the engine at a more appropriate time. In this sense, systems that seem "fully automated" can benefit from humans in the decision system. However, some systems are designed to prevent human intervention even at these broad levels.

3.3 AIRCRAFT CONTROL

The control of a modern, highly automated aircraft is an example of a semi-Structured decision process. Like temperature control, aircraft control is chosen for analysis because it is a highly evolved example of supervisory control that is also an *Unstructured into Structured* process topology. However, aircraft control is more complex, less subjective, and is dominated by issues such as safety.

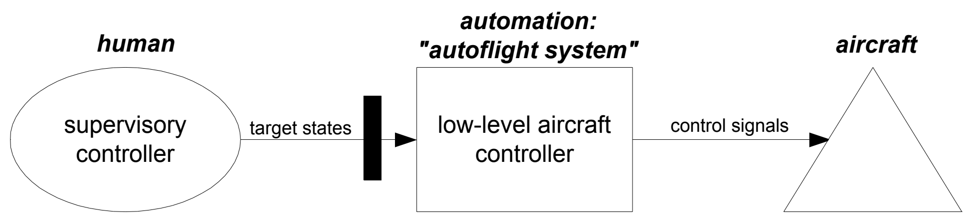


Figure 3-9 Simple model of aircraft control system

Figure 3-9 illustrates a simplified model of a two-level supervisory aircraft control system. As with temperature control, the human issues a well-defined target state—such as heading or altitude—which automation controls using well-defined rules. As is typical in supervisory control systems, the lower level is associated with shorter time and space horizons, allowing the human to provide commands less frequently.

Automating lower-level control tasks provides numerous benefits, but also introduces new problems. Since the flight crew has more time and attention available for other tasks, automation has allowed the crew size in some commercial aircraft to decrease from three to two, providing economic value. In fact, automated flight control can sometimes provide superior performance, such as optimum-fuel trajectories. However, while automation has generally improved the safety and efficiency of flight operations, new difficulties have emerged with human-automation interaction. Of particular concern is the loss of situation awareness and familiarity associated with higher levels of control and more-complex automation.

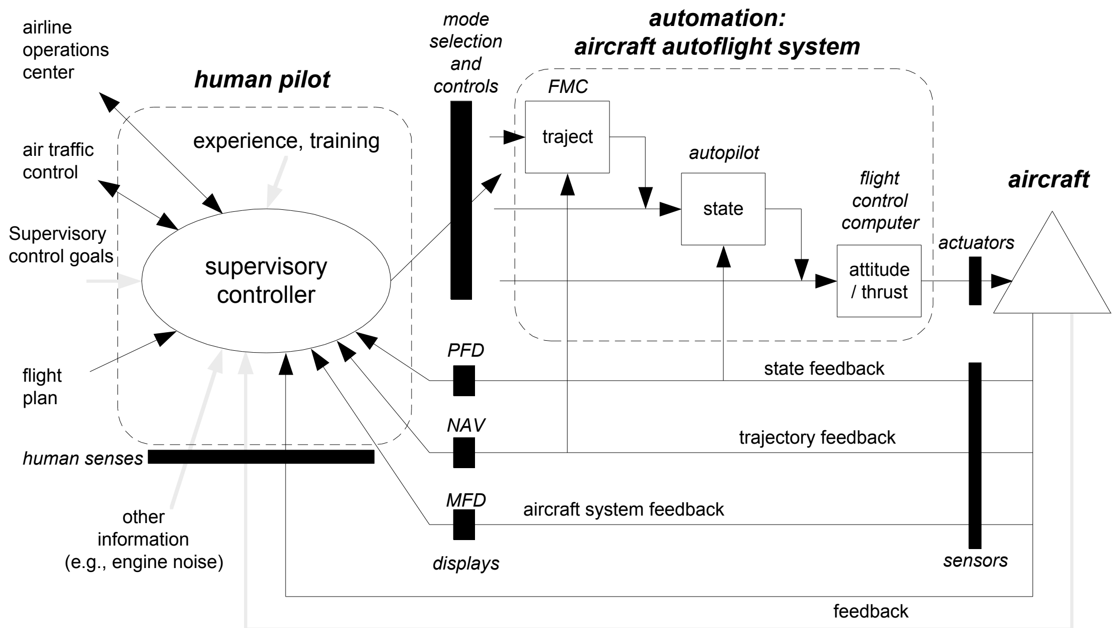


Figure 3-10 Detailed model of aircraft control system (multiple modes shown)

Figure 3-10 illustrates the aircraft decision system at a higher level of detail. Three primary modes of control are shown, any of which can be selected by the pilot. This strategy is known as dynamic allocation, since functions are allocated between humans and automation during operation, rather than during design. Despite that flight decisions are largely proceduralized [112], the pilot remains in charge of mode selection and all flight control decisions, and is ultimately responsible for passenger safety. These control decisions are supported with a variety of information from interfaces and natural senses. The following sections discuss cockpit decision making in both emergency and nominal flight conditions.

3.3.1 Cockpit Decisions During Emergencies

The most important function of the flight crew is to ensure passenger safety under all operational situations. Since designer's cannot account for all possible situations, humans are important in decision systems particularly during flight anomalies or emergencies, when prescribed processes no longer appropriate.

Responsibility and Control Authority

An important reason for having a human as a supervisory controller is that they can accept the *responsibility* for in-flight decisions. Responsibility is a way of propagating societal goals to a human decision-maker (represented as "experience" in Figure 3-10) such that he or she will make decisions that reflect the interest of others. Humans are believed to make better decisions than programmed rules during most unanticipated situations. Billings [15] cites part 91.3 of the Federal Aviation Regulations in this context:

1. *The pilot in command of an aircraft is directly responsible for, and is the final authority as to, the operation of the aircraft.*
2. *In an in-flight emergency requiring immediate action, the pilot in command may deviate from any rule of this part to the extent required to meet that emergency.*

In order to be responsible for decisions, the pilot needs to have control *authority* for decisions [88]. This is generally the case with aircraft. Other than the lowest flight control loops, which are essentially hardwired to the actuators, the pilot usually has the complete authority to intervene at any level. By at least providing the pilot with control authority and, hence, decision freedom, he or she is given the opportunity for responsible decision-making.

A responsible decision-maker needs to have an understanding of other people's goals in order to act in their best interest, and to understand the longer-term social consequences of ignoring these interests. The understanding of goals is perhaps the most logical argument for giving human pilots control authority. For example, there is little question that a human pilot will access whatever resources available to preserve the safety of the passengers because humans share common Goals—in particular the instinct to *survive*. In fact, one can argue that the pilot can even act selfishly for his or her own survival because decision consequences (in terms of human safety) apply equivalently to everyone on board. In any case, the Goal of survival is an inherent human trait that cannot be applied to machines, providing justification for the belief that a human pilot will act responsibly to ensure a safe flight.

Adaptability

During flight emergencies, it is also believed that humans can use resources effectively to make situation-specific decisions that cannot be preprogrammed. In other words, humans appear to be able to effectively *adapt* to unanticipated events.

Although some researchers debate whether humans are, in fact, effective at adapting to truly novel situations [121], there is a general consensus that humans are more flexible and adaptive than their machine counterparts [14], [138]. Billings [15] states that "…pilots and air traffic controllers are essential because they are able to make good decisions in difficult situations. We have not yet devised a computer that can cope with the variability inherent in the flight and air traffic environment." This view is similar to that of Fitts over fifty years ago [43].

The adaptability of humans can certainly be attributed, in part, to their broad experience in professional training and, more generally in life. The breadth of knowledge from experience is an important resource for adapting to new problems. Experts such as pilots are often able to easily detect differences from their expectations (recognize a problem exists), and formulate novel strategies based on analogies from previously-solved problems [72].

As in the case with responsibility, the ability to adapt can be positively influenced by an understanding of the system goals. For unanticipated situations, it is good that pilots do not follow a scripted set of rules. The ability to deviate from these rules, to deal with contingencies in whatever manner seems fit, is a valuable human resource.

At the very least, the Unstructured process allows humans to access information during operation that extend beyond the inputs that are designed into Structured processes. Consider the inputs to the Unstructured process in Figure 3-10. Humans can view, through displays, any information that is accessible to automation—such as from sensors and databases. *In addition*, humans perceive information that is naturally sensed. This provides an additional, sometimes redundant, source of information that can be critical for both recognizing that a problem exists, and for determining corrective actions. For example, a pilot may be able to diagnose a problem from the complex pattern of aircraft symptoms that are unique to a particular failure—in part because they do not have to understand the recognition process. "Other information" (Figure 3-10) can be incorporated during operation, even if actively sought, without a prescribed process for obtaining it. Since an implication of rule-based decisions is that their inputs are prescribed prior to flight, Unstructured decisions have an advantage in their ability to access information that is not anticipated. In this informational sense, the belief that humans can effectively adapt to aircraft emergencies can be reasonably justified.

Closing Remarks on Flight Emergencies

Since safety is a primary driver in aircraft decision systems, humans appear to be valuable primarily for their responsibility and adaptability during anomalous in-flight emergencies. Both attributes are strongly influenced by an understanding of the system goals during emergencies. In addition, the ability to adapt can be justified by the additional information that humans can access.

It is relevant to ask whether the above traits are unique to humans, or to Unstructured decision processes. The concept of responsibility and the understanding of goals may provide a reasonable basis for arguing against rule-based machines as a supervisory controller. However, "human error" has been identified as the reason behind many accidents with complex automation [61]. As Reason [121] has argued, a "catch-22" in the control of complex systems is that, while humans are often placed in systems for their adaptive abilities, "each incident is a truly novel event in which past experience counts for little." For some situations, the value of humans is merely perceived: human operators make people *feel* safe and comfortable [15], [138].

Neural networks are subject to similar limitations as humans with respect to experience in novel situations. That is, they are not likely to effectively adapt to a single event in which a reasonable database for training does not exist. "Common" events such as engine failures may have available data for training neural networks in diagnosis and response, but humans, too, are trained for these emergencies. Hence, neural networks may not offer any clear advantage in novel decision-making. However, as aircraft systems become more complex, even anticipated failure modes may become increasingly difficult for operators to identify, in which case neural networks and traditional rule-based algorithms can be valuable.

3.3.2 The Aircraft Control Hierarchy

As supervisory controllers of a complex system, the flight crew issues control commands and monitors automated tasks. Automation can be selected to operate in a variety of control modes, which here represent distinct levels in a control hierarchy. In this section, the decision processes associated with the different control levels are analyzed in order to understand the requirements of the highest level: the human supervisory controller.

A Spectrum of Structured Controllers

It is useful to view control automation as a spectrum of Structure, as shown in Figure 3-11. Although the criteria for a "degree of automation" is not precisely defined, the five instantiations in Figure 3-11 serve as discrete levels within a continuous spectrum. In this diagram, more Structure (automation) is represented as a larger rectangle—and hence a correspondingly smaller oval (human). As the automation increases, the function of the semi-Structured process remains unchanged (same system function), but the role of the human supervisor shifts from a controller to more of a *manager* of automation.

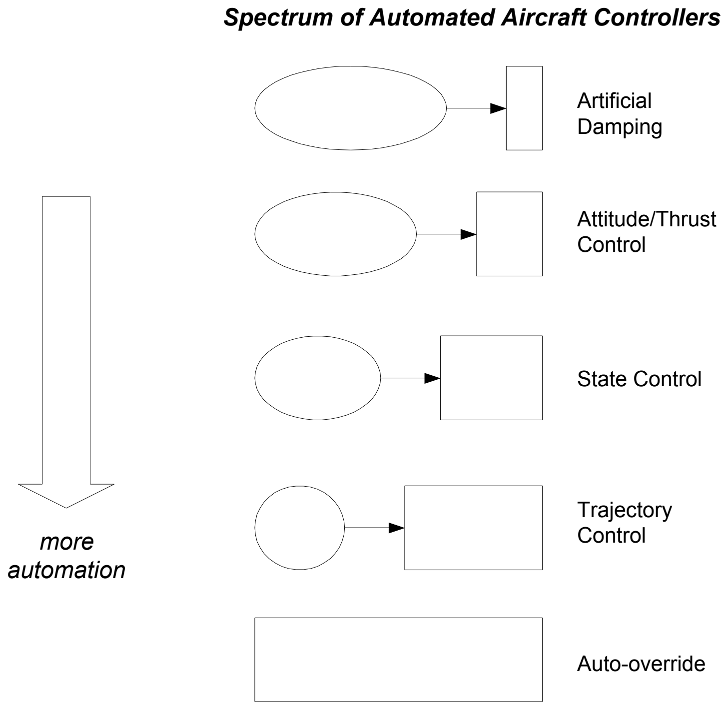


Figure 3-11 Spectrum of semi-Structured aircraft controllers

The lowest level of automation in Figure 3-11 is *artificial damping*, in which automation augments human control inputs to attenuate certain frequencies or to augment stability. This is often a permanent “hardwired” aircraft element, as is *attitude/thrust* control. In the latter, control inputs are translated to actuator states, such as control surface positions or thrust, sometimes with closed-loop capability. At the *state* level, automation controls states that are associated primarily with the flight path—heading, sink rate, altitude, etc.—based on mathematical integration of the lower level attitude/thrust states. At the *trajectory* level, automation coordinates state modes for piecing together a flight trajectory. Finally, although not common, the *auto-override* function allows automation to seize control of the aircraft (usually for safety reasons), based on well-defined criteria.

Analysis of Attitude, State, and Trajectory Controllers

The following paragraphs discuss three of the five levels of automation shown in the control spectrum of Figure 3-11. These three levels—attitude/thrust, state, and trajectory control—represent selectable modes in the detailed diagram of Figure 3-12. (This three-tier representation is a simplification, since aircraft often have dozens of modes within each level). Each controller is selectable by the pilot, and can mutually interact *hierarchically* to form the aircraft "autoflight" system. That is, the trajectory controller provides goals to the state controller (e.g., heading), which can then provide goals to the attitude/thrust controller (e.g., roll), which produces actuator commands. The hierarchy is illustrated in Figure 3-12 as nested control loops.

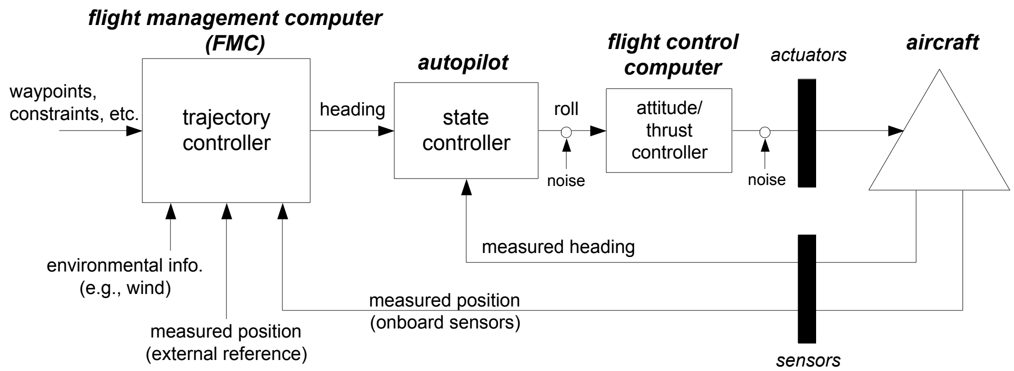


Figure 3-12 Control hierarchy of the autoflight system

Attitude Control

At the attitude level of control, which was one of the first forms of automated flight control, the rules are designed using standard techniques from control theory. The mapping from an attitude command to an actuator state is based on a set of differential equations that represent the aircraft dynamics. These equations, along with a model of environmental noise and disturbances (e.g., air turbulence), provide a complete framework for control analysis and the evaluation of performance (e.g., stability, robustness, response time). Feedback is not necessarily required since control actions are primitive and well understood at this level. Hence, a *single* control input—an attitude goal—may be sufficient for attitude control decisions or outputs: the signals for actuators.

State Control

At the state level of control, the design techniques are similar to those at the attitude level. Again, control theory provides the necessary tools for decision rules, based on models of the aircraft dynamics. However, states such as heading or altitude are higher level and more general than attitude, since they can be calculated from the integration of attitude and thrust over time. Also, there is a greater dependence on feedback, since the uncertainties in the mathematical model accumulate errors over time, particularly as the result of integration. Uncertainties in the environment are still considered noise—as is the case with attitude control—but the controller is robust to this noise by using sensor feedback that measures states, such as with air velocity and radar altimetry. Hence, state commands (target states) and sensor feedback (measured states) are the two primary pieces of information required by the controller; other information is considered noise or disturbance. From these two inputs, state control decisions calculate an error that is then minimized in some sense over the course of that trajectory.

Trajectory Control

The trajectory controller, which is the highest automated control mode in Figure 3-12, differs from state and attitude controllers. Its function is more of a manager than a controller: to coordinate a series of target states in order to construct a trajectory that satisfies the primary flight goals. Target states, when achieved, serve as transitions to new target states, continually pulling the aircraft towards its final destination by means of the state controller. Hence, target states at the trajectory level exist merely to decompose a trajectory into a series of temporary sub-goals that are simpler to articulate, control, and monitor. The trajectory controller is not like standard feedback controllers, whose goal is often to *continually* drive the target state error to zero.

For background, a typical scenario for trajectory control is to translate a series of waypoints into state control targets. The combination of radio ground beacons and inertial sensors provides three-dimensional measurements that are sufficiently accurate for defining targets as positions. This has allowed automation to reliably execute a sequence of flight paths based on well-defined criteria. Furthermore, automation can factor wind, time constraints, and other goals to calculate the most efficient state profiles for a particular aircraft. Hence, waypoints can also be considered trajectory *constraints*, in which the goals are to optimize fuel/time efficiency within these constraints.

Trajectory control differs from lower levels of control in the amount and type of information required for the decision. In addition to the target states (waypoints) and present states (measured position), the trajectory control decision uses additional information . Much of this additional information characterizes the environment for *planning* the state profile: the variation of altitude, speed, etc. that achieves the optimum performance. Information such as wind replaces what would be considered "noise" in the lower levels. Here environmental information is obtained during operation and explicitly factored into the decision.

Lastly, a trajectory controller needs to anticipate future states in addition to its more immediate states. For example, when climbing to a certain altitude the trajectory controller anticipates what lies ahead, and begins to transition before the target altitude is reached. As with the anticipation of environmental features like wind, the anticipation of longer-term goals reflects the strategic versus tactical nature of trajectory decisions.

Summary: The Autoflight Control Hierarchy

The three modes of automation analyzed in this section—attitude, state, and trajectory control—illustrate a pattern associated with the level within a hierarchical control system. Lower levels are simpler—in terms of their goals and information requirements—and theory exists from which to generate the rules for decision-making. The information required for lower-level decision-making does not include measurements of the environment; these uncertainties are accommodated with feedback and robustness [161]. As the trajectory controller illustrates, higher levels require not only *feedback* of present information, but additional information for planning and management—information for *feedforward* control. Simply, analysis of automation shows that decisions at higher levels are more complex.

The Human Supervisory Controller

The previous section indicated a trend in complexity associated with higher levels within an automated controller hierarchy. This trend can be extended to help explain why humans—more importantly, why Unstructured processes—are at the highest level: the supervisory controller.

Consider the generation of waypoints for trajectory control. As might be expected, the pilot has some freedom to determine these during flight: to shape the trajectory based on new information that is acquired about the environment. This is an important ingredient for strategic decision making. Humans observe air traffic, weather systems, terrain maps, and other information (Figure 3-13). Some of this may be perceived directly (e.g., through window), but the majority of information for navigation is supplied through the Flight Management System (FMS) via an interface. Although automation has access to the same information, the manner in which it is incorporated into a decision is ill-defined, and hence inappropriate for rules. Therefore, humans may be able to incorporate more information than automated controllers—a characteristic that is consistent with the observations of levels within the autoflight system.

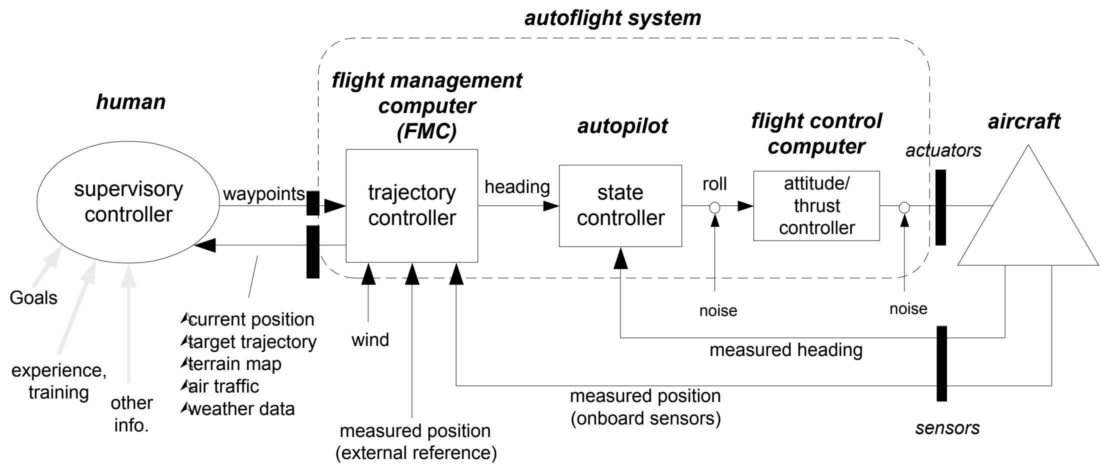


Figure 3-13 Humans are at the highest level in the aircraft control hierarchy

The decision to alter the flight trajectory is ill-defined for multiple reasons. It is believed that experienced pilots rely strongly on their judgment and intuition to determine the necessary trajectory adjustments. Dangerous situations are recognized, and judgments of safety are made, based on the interpretation of weather systems and other potential hazards—particularly in the presence of uncertainty. At the same time, tradeoffs are made based on the goals of the airline and passengers regarding fuel use and arrival delays. Training and experience are valuable resources for such judgments, as well as the perception of complex patterns of information.

Another important attribute of humans is their knowledge of the world [14]. The information that is gathered during flight is interpreted in the context of a "bigger picture" of which experience has provided [135]. This essentially is equivalent to observing additional information about the environment, and seeing further into the future, both of which have been shown to be characteristics of the strategic planning required by high-level decision-making. Knowledge of the world is also important in understanding when the environment is appropriate for automation, and when human intervention is necessary.

Closing Remarks on the Aircraft Control Hierarchy

Humans appear to be valuable as supervisory controllers because they can accommodate the requirements associated with high-level control. In the previous section, it was observed that higher-levels of control appears to require:

- more-complex information
- more-complex goals

- information about the environment, opposed to just the controlled system
- information that supports long term *strategic* versus *tactical* decisions

These characteristics are also associated with human supervisory decisions in aircraft control, sometimes in an extreme sense. The human's Unstructured decision process can perhaps be viewed as the decision process that emerges as an "upper limit" in a hierarchical control system.

Humans complement automation by their ability to understand the bigger picture associated with the information observed during flight. Their knowledge of the world acts as a supplement to displayed data, and a buffer to missing data. They can understand when the environment is appropriate for automation, and can make periodic adjustments using simple well-defined parameters to extend automation's envelope of operation. It appears that many of the decisions made by humans as a supervisory controller are ill-defined, so that Unstructured processes add value to the high level control of complex systems.

3.3.3 Displays for Aircraft Control: Informing the Human

As a supervisory controller of a complex system, humans benefit from information to support the multiple levels of control and management. Some of this information is sensed naturally, without the aid of automation. Other information is obtained from designed information channels through human-automation *interfaces*. In this section, information obtained through interfaces is analyzed and compared for the three modes of control.

Display for Attitude and State Control

The human-automation interface for observing information for attitude and state control is assumed to be a "glass" (cathode ray tube) monitor known as the "primary flight display" (PFD). As discussed in the previous section, these two levels of control are similar in their time and space horizons. The integration of multiple parameters into the PFD (older aircraft are through individual electromechanical instruments) facilitates visual scanning of control information that is frequently accessed.

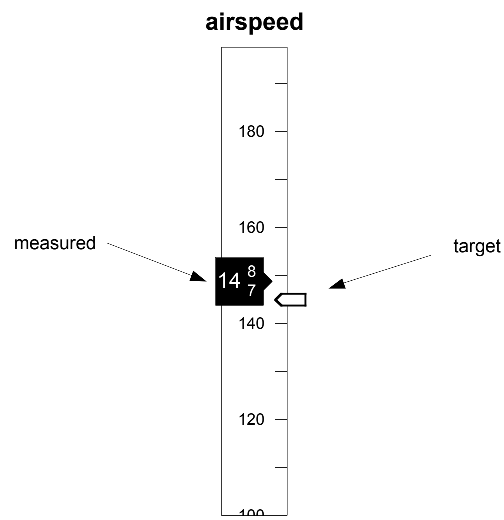


Figure 3-14 Air speed display: one of many in the primary flight display (PFD) that indicates both target and measured states

The information provided by the PFD for inputs to humans is nearly identical to the information provided to automation (for attitude and state control). Attitude feedback parameters are roll, pitch, and yaw, and state feedback parameters are airspeed, altitude, and vertical speed. All six feedback elements are displayed in an unambiguous form relative to a target state and/or other important reference points. An example state within the PFD is airspeed, shown in Figure 3-14. Note the similarities to the traditional temperature "dial" thermostat from Figure 3-6. Two parameters—the target state and measured feedback—on a calibrated scale appear fundamental to feedback control. Here, they provide both absolute and relative indications of the aircraft's current state and future state, with the advantages of both analog and digital formats. In addition to PFD information, humans also sense commands and state feedback naturally, providing potentially new information, redundancy, and/or an added dimension of meaning.

Due to the complexity of highly automated aircraft, it has also become important to provide information pertaining to the states of *automation,* in addition to the states of the *aircraft*. This is accomplished through an indication of the *mode* of the autoflight system. Autopilots, for instance, often have a mode annunciation panel to display to the flight crew a clear indication of these states. Mode awareness has been identified as a factor in recent accidents involving modern aircraft [61], particularly since the number of possible modes and sub-modes is large.

In summary, the PFD is a display that provides the flight crew with three types of information. Two of these—the target state and the current measured state—are provided for each of the six control variables (although target states for attitude are fed back through stick positions: a separate interface). This information pair is common to automated feedback controllers in general, and is the case with attitude and state controllers. In addition, the mode of automation is another state that the interface displays—one that is not provided in simpler systems. This information set is shown in Figure 3-15, which is simplified to show generic display information. The three pieces of information provide the operator with a sufficient sense of the big picture: the current aircraft state, the future state, and, with mode feedback, an indication how this state transition will occur.

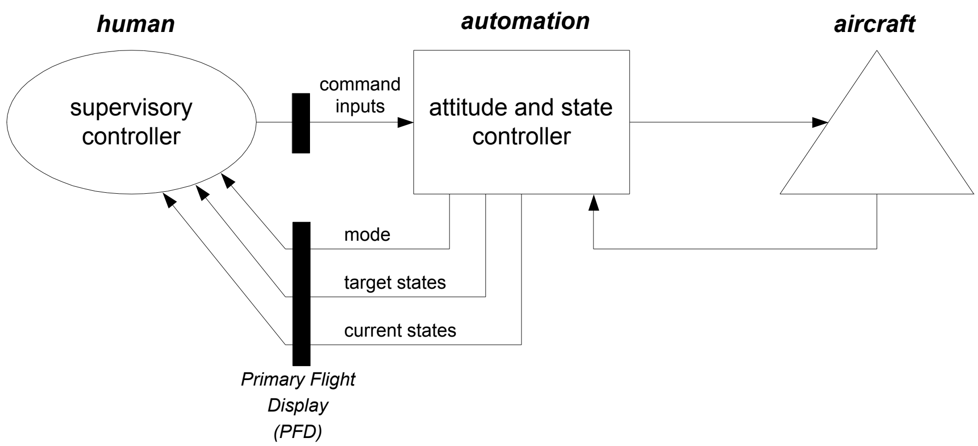


Figure 3-15 Information designed to support attitude and state decisions

Display for Trajectory Control

The information provided to the human for trajectory control builds upon the basic information set used in attitude and state control. As mentioned earlier, trajectory control is more of a management function than a control function, and its strategic requirements demand a more complex information set, particularly for long-term planning. Automation has been critical in integrating many sources information into a single graphical display, referred to as the NAV (Figure 3-16). For horizontal navigation, this display provides an intuitive planview map of the local environment, and allows pilots to visualize trajectories, even before they are officially entered.

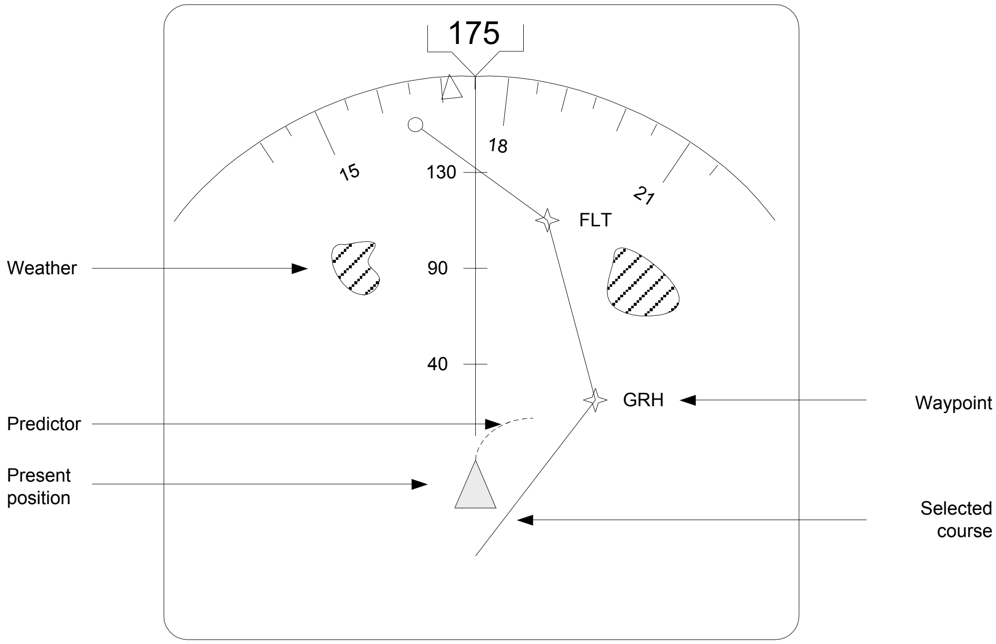


Figure 3-16 Navigation display (NAV): the interface for trajectory control

The NAV provides a graphical display of the aircraft's current position relative to its target states: the programmed trajectory. Predictors can also be used to indicate how state transitions are expected to occur. This information set is similar to simpler controllers, as discussed in the previous section with state and attitude control. In this case, the "big picture" requirement associated with trajectory control is literal, and in fact the image in the NAV can be zoomed to meet the dynamic needs of the operator.

The NAV also displays potential hazards, such as weather systems from radar, terrain maps from an onboard database, and air traffic from collision avoidance systems. As mentioned, these are features of the environment that cannot be treated as noise such as light turbulence; they need to be observed well in advance and avoided. The NAV provides a way to quickly visualize this information.

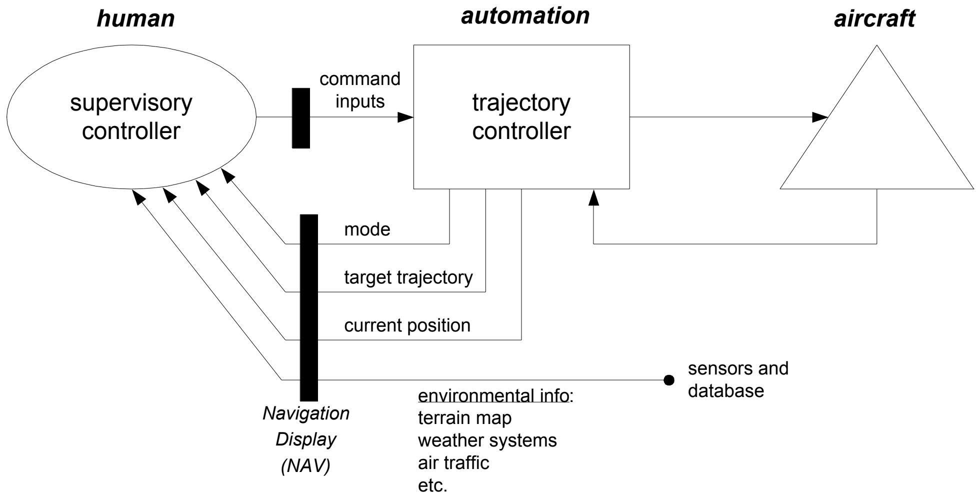


Figure 3-17 Information designed to support trajectory control decisions, which includes environmental information

Figure 3-17 shows the generic information flow for supervising an automated trajectory controller. Note that, although the information content is different than in state and attitude control, the diagram is similar in a generic sense. Mode, target states, and current states are displayed. However, an important difference is the addition of environmental information to the NAV display, which the human incorporates in an ill-defined manner. This information has been identified to make navigation decisions more informed.

3.4 AIRCRAFT COLLISION AVOIDANCE

This section describes the use of automation as front-end processing for informing human decisions. The resulting decision system has a *Structured into Unstructured* process topology. "*Information* automation"—in which computers are used to generate inputs to humans rather than to process human outputs—is similar to control automation in that it has different "levels," which here includes data filters, graphical displays, alerts, and decision aids. These are sometimes generically referred to as "observers" instead of "controllers." An example from aircraft collision avoidance is used to illustrate some of these functions—in particular alerts and decision aids. Figure 3-18 illustrates information automation for aircraft control (control automation is not shown).

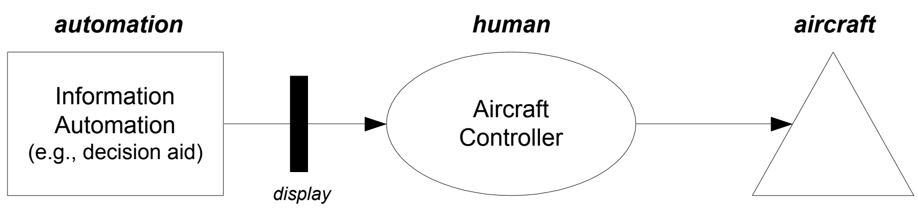


Figure 3-18 Simple model of information automation in aircraft control

Assuming that a human is involved in a decision, there is value to processing information with well-defined rules for providing inputs. Simply put, the purpose of Structure in this situation is to make the human *informed*. Informing humans—increasing their "situation awareness" [39]—means more than simply providing information about anything he or she may possibly want to know, since humans have a limited capacity for serial processing and attention [140]. Informing the human can mean different things for different decision-makers, tasks, environments, workloads, etc. However, when these issues are sufficiently understood, Structure is often designed into decision systems as a prescribed front-end process.

Information automation performs work that is analogous to human subordinates in managerial decision-making (this analogy is also used in supervisory control). Managers rely on their staff to perform cognitive work for information tasks, opposed to physical work for control tasks. Employees filter phone calls, provide executive summaries, take minutes, interrupt meetings during emergencies, and provide expert advice. Managers then use their judgment and incorporate this “processed” information appropriately. Similarly, *automation* performs information tasks using prescribed rules and procedures, which the human decision-maker then uses in an ill-defined manner.

3.4.1 Spectrum of Structure

It is useful to consider information automation on a spectrum. Figure 3-19 shows five instantiations along this spectrum, along with a similar diagram from control automation (from Figure 3-11). For comparison, automation used in this manner can be considered “observers” opposed to “controllers.” Note the semi-Structured symmetry as automation is increased top to bottom.

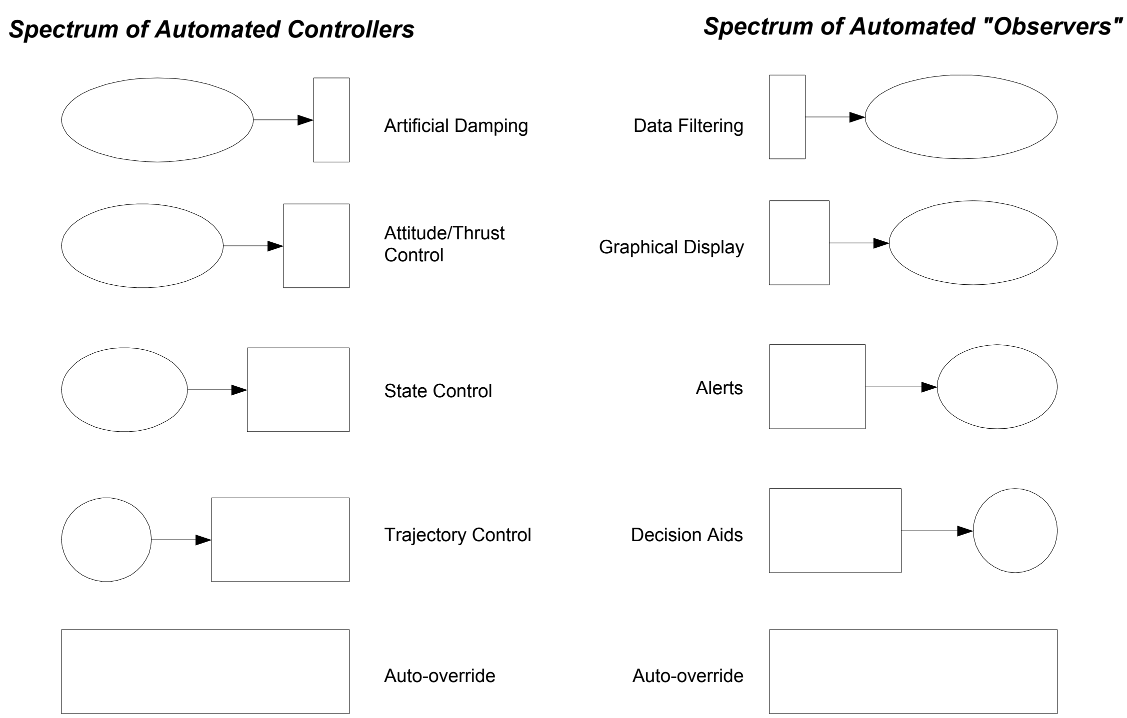


Figure 3-19 Spectrum of semi-Structured controllers and “observers” (information automation) in aircraft decision processes

The progression within the right column in Figure 3-19 is based on the function of automation within the decision system. At one extreme (top), functions such as filtering provide minor processing, and can even be considered part of the interface. Near the other extreme, functions such as decision aids can potentially be used as a replacement for humans, which is in fact the case when automation *overrides* humans, temporarily removing them from the decision system. Note that both columns culminate in auto-override.

Filtering

Filters are an example of a low level of Structured observers. Consider reading a digital value from a DC voltmeter. Anyone who has recorded such data has likely encountered difficulties when the numerical value fluctuates. When fluctuations are slow enough to read successive values, people often make an estimate of the DC value based on a "mental averaging" of some sort. A similar mental averaging might be used when people weigh produce at a market, since the scale mechanism often damps slowly. In both cases, the process of reducing a time sequence of data into a single number can be reduced to well-defined rules, such as a simple mathematical average. Figure 3-20 shows two decision process, one of which a the Structured averaging process is allocated to automation (Figure 3-20(b)). In Figure 3-20(a), the human is performing a similar filtering process in an unknown manner. The rules in Figure 3-20(b) may be valuable for their repeatability, accuracy, speed, or perhaps simply to save humans from the mental work of averaging.

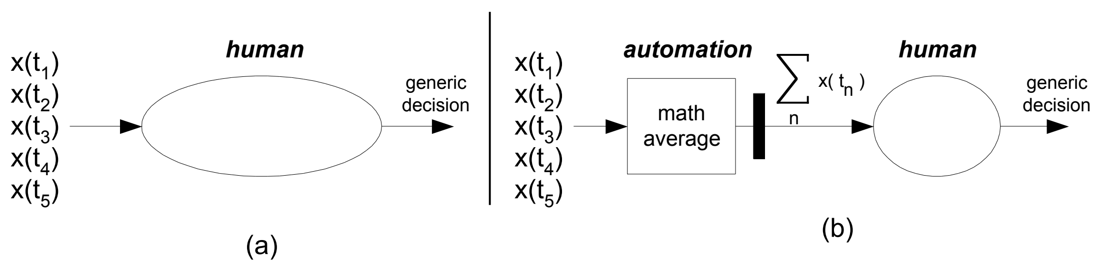


Figure 3-20 Supporting human decisions by automating a filtering process

Graphical Transformations

The representation of information can affect a decision process. Graphics are a particular type of representation that allows people to use their natural perceptive abilities to see features or patterns differently from symbolic representations. The relevance here is that graphical transformations are almost exclusively through a Structured process. For example, the spatial mapping of numerical data to a position on a plot is through formal rules.

It is clear that a Structured graphical process can add value to human decision-making, particularly when the process is automated. New representations can provide a deeper insight into patterns and relationships, or be better suited to a particular task [106]. Sometimes, as with filtering, a formal process may be a substitute for an existing mental transformation. For example, Dehaene [29] has found that people describe *numbers* as having particular shapes, colors, etc., and that humans appear to have an innate (but approximate) mental "number line." In any case, human decisions (Figure 3-21(a)) are often better-informed with a Structured graphical transformation, as in Figure 3-21(b). Of course, automated graphics require a human-automation graphical interface.

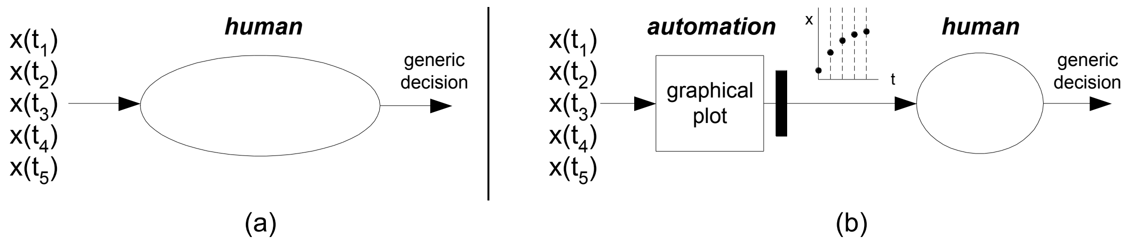


Figure 3-21 Supporting human decisions by automating a graphical transformation process

It should be mentioned that humans do not *necessarily* process graphical information in an ill-defined manner. For example, a simple control task may be to change gears when the RPM needle passes the vertical position. In this case, the operator uses a rule: "IF needle passes threshold, THEN change gears." It is not even necessary that the operator understands what the needle position represents—only that its position serves as a conditional input to a rule. Graphical information used without an attached meaning is called a "signal" or "sign" [120]. This technique can be used to easily solve problems in a certain representation. For example, the optimization of a scalar function can be to simply find the highest point on a curve. As Simon [141] has noted, a well-represented problem is almost nearly solved. *Graphical* representations often provide an intuitive representation for decision-making, even when the decision process is Structured.

A more interesting issue is how people use Structured graphical transformations of information as inputs to Unstructured decisions. The representation of information is known to affect the decision process [106], [130], and this characteristic is sometimes exploited to elicit perceptual, intuitive decisions [53]. In data exploration, patterns may be discovered unpredictably. It is difficult to understand precisely how this mechanism occurs, particularly prior to data analysis. Even when displays are fixed, as in an automobile speedometer, Structure is imposed by the system designers without a full understanding of how this representation will affect operational decisions.

Alerts and Decision Aids

Alerts and decision aids are further along in the spectrum of Figure 3-19. These functions will be discussed in depth for a specific application in the next section. Here, alerts and decision aids are briefly discussed in a more generic sense.

Alerts

Just as filters and graphics transform information with formal rule-based processes, *alerts* also operate on well-defined information to yield a new representation that adds value to human decisions. Typically, the new representation is of the form of an unambiguous hazard state, through interfaces such as warning lights or aural tones. Multi-state alerts provide added resolution of hazards. Their primary benefit is to call attention to a situation that may otherwise not be recognized. Failure to recognize or have sufficient situation awareness can be due to complexity (e.g., the fusion of multiple data sources), unobservable data, or workload/attention. Although issues such trust and false alarms can be problematic [138], alerts generally improve a decision-maker's situation awareness about a hazard. However, alerts do not provide information about how to resolve a hazard.

Decision Aids

Decision aids use observed data to propose one or more decision options to a human. Computer-generated advice can apply towards any situation, including providing options for resolving hazards, as in the following aircraft example.

Decision aids typically generate advice using standard production rules. Aids can be of particular benefit when a decision-maker is not in the position to make good decisions, whether due to inexperience, time pressure, or other factors. Decision aids can also reduce the effort involved in searching for alternatives, allowing humans to focus on a more manageable set.

In general, however, decision aids are only used as *aids* because humans are believed to add value to rule-based decisions, which operate on a fixed set of information in a context-free manner. In contrast, humans are able to incorporate additional information, ignore information, interpret the operational context, use creativity, apply judgment, etc. It should be mentioned that decision aids have also been known de-skill humans or cause misplaced trust [138]. In these cases, humans sometimes use decision aids in a Structured manner—such as by selecting the "best" option as determined by the computer. Typically, though, the manner in which humans incorporate advice into decisions is often ill-defined, resulting in a decision system that is semi-Structured, as in Figure 3-18.

Summary of Information Automation Spectrum

Filtering, graphical transformations, alerts, and decision aids are four examples of Structured processes that are used "upstream" in the information path, providing inputs to human decision processes. People generally do not *require* these inputs, for they can rely on other resources—including their experience and ability to access a different set of information than that designed into automation. However, these automated processes are designed to improve decisions by making a person's decision process more informed.

3.4.2 Aircraft Collision Avoidance: Ground Proximity Warning System

This section analyzes a specific system for illustrating alerts and decision aids: aircraft Ground Proximity Warning System (GPWS). A detailed diagram is shown in Figure 3-22, which identifies relevant information flow and interfaces.

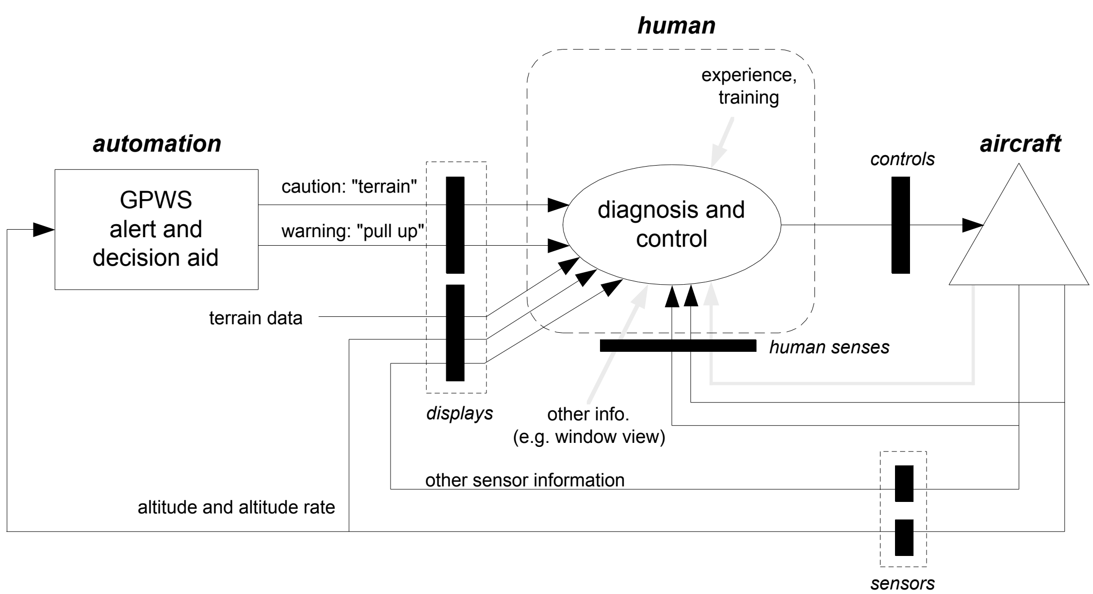


Figure 3-22 Detailed aircraft collision avoidance decision process

The GPWS was introduced in 1974 to prevent "controlled flight into terrain" accidents for civil transports. It consists of an alerting and decision aid system that informs the flight crew of a potentially hazardous situation, based on aircraft sensor measurements such as radar altimetry. Sensor information—primarily altitude and altitude rate—is processed by automation using well-defined rules, which then leads to alerts based on fixed thresholds. The alert is, in practice, two-stage, providing "cautions" for the first stage, and "warnings" for the second—the latter of which is accompanied by decision aid advice for hazard resolution. GPWS is designed to improve the flight crew's situation awareness about the environment, and to provide advice if immediate action is required. Both inputs support human decisions, adding to the rich set of information that can be used during operation.

Alert Logic

The rules for GPWS alerts are a function of its altitude and altitude rate. Other information (e.g., aircraft configuration) is also used, but these are ignored here for simplicity. Altitude information comes from two sources:

1. *barometric altitude,* h_b – derived from air pressure
2. *radar "above ground level" altitude,* h_r – derived from the time-of-flight of electromagnetic pulses between aircraft and terrain

Altitude rates, $\dot{h}_b$ and $\dot{h}_r$, are also available. Since GPWS uses primarily altitude information, and does not observe lateral states or intended path information. Figure 3-23 illustrates the observable states. Despite this somewhat limited observability, GPWS has played a part in reducing "controlled flight into terrain" accidents [11].

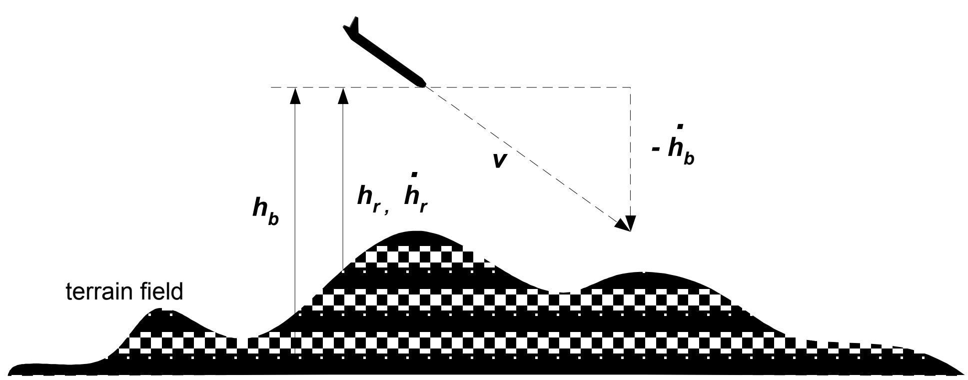


Figure 3-23 Observable states used by GPWS alerting logic (based on Kuchar & Hansman [78])

Since measurements are downward-looking, alerts are based on assumptions of terrain variations. Radar measurements from directly below the aircraft are extrapolated to project the terrain ahead of the aircraft, in the direction of its flight. This projection is required to provide advanced warnings to the flight crew due to the momentum of the aircraft, and latency in their response. The tunnel vision of GPWS is prone to errors—false alarms and missed detections—depending on the terrain profile, as in Figure 3-24 (a) and (b), respectively. Since these errors are generally in conflict, they form a common tradeoff in threshold-based alerts.

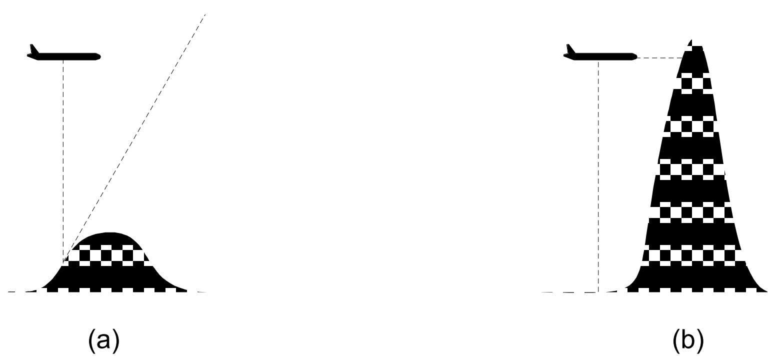


Figure 3-24 Illustration detection errors. In (a), alert logic based on terrain closure rate produces false alarms, opposed to missed detections in (b) (based on Kuchar & Hansman [78])

The alerting logic depends on the aircraft: more responsive aircraft generally require less warning time. Combinations of altitude and altitude rate (or terrain closure rate) define a two-dimensional state space in which near-hazardous regions can be defined. Intuitively, dangerous regions in the state space should exist at low altitudes, high barometric descent rates, and/or high terrain closure rates. Figure 3-25 shows example alerting threshold diagrams for a *Boeing 767* aircraft (Boeing 1983). These diagrams, which define the alert space, are similar, but based on different hazard assumptions. Each defines two main regions that define the severity of the situation: a *caution* region (light gray) and a *warning* region (dark gray). The caution regions are considered *alerts*: they call attention to the situation ("sink rate" or "terrain"). The warning regions are also *decision aids*: they provide advice on how to resolve a hazard ("pull up").

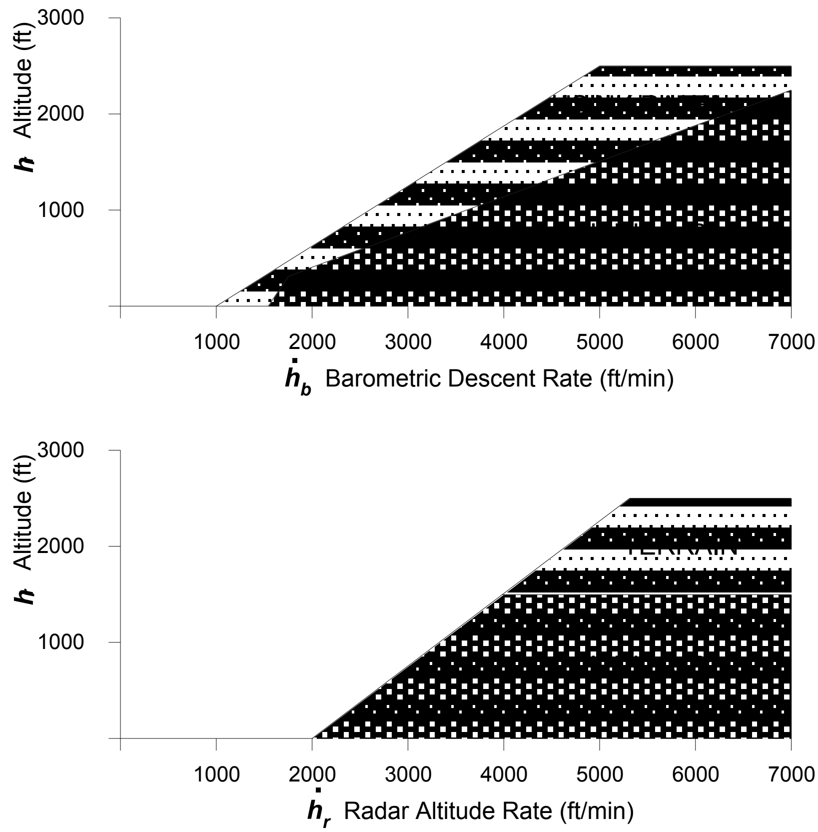


Figure 3-25 Approximate alert and decision-aid logic for *Boeing 767* GPWS system (based on Kuchar & Hansman [78])

Both alert threshold diagrams illustrate the Structured process that automation uses to inform the flight crew. Alerting logic is based on a simple, well-defined set of information. This logic is not likely to be identical to what human operators use. However, given the contraints of well-defined rules, GPWS provides value to a broader, more-complex decision process, in which humans are actively involved.

Human-Automation Interface

Displays for alerts are both visual and aural. Amber lights are used for cautions, and red lights for warnings. These lights include descriptive information: “GND PROX” for cautions, and “PULL UP” for warnings. These are accompanied by aural descriptions: “Sink rate” or “Terrain” for cautions, and “Pull up” for warnings. Interface information is summarized in Table 1. Note that cautions describe reasoning for the alert, and warnings describe actions for resolving the crisis. If the situations were not time-critical, warnings perhaps may have also included information about the nature of the hazard.

Table 1 Visual and aural information from the GPWS interface

	Visual Indicator	Visual Info.	Aural Info.
Cautions	Amber light	“GND PROX”	“Sink rate” or “Terrain”
Warnings	Red light	“PULL UP”	“Pull up”

Human Response to Alerts

Upon receiving an alert or decision aid, members of the flight crew often resort to an ill-defined process as part of a more complex situation assessment and flight control decision (Figure 3-22). These processes are examined to understand the value that humans add to the decision system.

Humans and automation comprise a semi-Structured decision system that is synergistic for collision hazard diagnosis and control decisions. GPWS alerts provide value to humans by calling attention to hazards that may not be recognized by the flight crew, particularly during poor visibility. Situation awareness tends to degrade particularly during periods of very high or very low workload, for reasons such as limited attention resources, fatigue, boredom, complacency, etc [138]. GPWS is clearly valuable, but it may not be sufficient for reliable situation assessment due to errors previously discussed.

Ground Proximity Detection

One of the ways humans add value to ground proximity detection is through their access to more information. Whereas GPWS alerts use sensor measurements from views directly beneath the aircraft, humans can often observe images whose field-of-view covers laterally—in particular the forward direction. This information may not be as precise as artificial sensors, but it provides broad information about the “big picture”: the current state of the aircraft relative to where it has been and where it is going. This information can be especially important for verifying alerts, since good visibility can be sufficient for almost instant confirmation or falsification of a terrain hazard. The flight crew can also access on board navigation maps for obtaining a similar broad terrain view. In addition, humans have access to the same sensor information as GPWS. As Figure 3-22 shows, humans have access to a rich set of information, including GPWS and its supporting information, direct visual images, and an array of standard flight control feedback.

Humans rely on experience also. Familiar trajectories can build an expectation of terrain, so that the flight crew can be more vigilant in dangerous areas, perhaps even accommodating missed detections. Similarly, they can learn to predict regions of false alarms. Experience can help to understand and predict GPWS behavior, which can potentially improve pilot conformance and trust in automation [117], [138].

Flight Response Decisions

The pilot response to a GPWS alert depends on its severity. During a *caution* ("sink rate" or "terrain" in Figure 3-25), the pilot is at liberty to devote resources to assessing the potential problem. This is because the alert space for cautions is conservative, allowing time for verification even in the event of a correct detection. However, the pilot is instructed to immediately perform a wings-level pull-up if visual contact with the terrain is not made when GPWS issues a *warning*: the more severe of the two-stage alert [28]. Automation augments this training with decision aid information: a visual and aural "pull up" instruction.

Despite training and decision aids, it has been observed that pilots do not always conform to GPWS warnings, leading to delayed or variable flight responses. In early GPWS systems with simpler logic, the high rate of false alarms resulted in pilots ignoring or disabling alerts—some resulting in accidents [15]. False alarm rates continue to be a factor, even in modern GPWS. Pilot responses and their motivation are not fully understood: the decision process appears Unstructured despite that training calls for a Structured response (procedure). Hence, in addition to hazard *diagnosis*, it appears that pilot *control response* to GPWS alerts can also be ill-defined.

There are potentially many contributing factors to Unstructured response behavior. These include a lack of trust in automation—which is affected by previous experience of false alarms—and/or a mismatch between the automation and the pilot's current expectation [117]. As with many alert response decisions, the decision to pull up seems to consider the consequences associated with automation error: the positive consequence of not conforming in the event of a false alarm. This may at first seem irrational, considering the grave potential consequences. However, studies have shown that human behavior under high risk is not modeled well by classical decision theory [31].

The concept of an Unstructured process not only provides a framework with which to view response decisions to GPWS alerts, but also helps to understand why it is difficult to improve pilot conformance. For example, one solution is to provide more information about a GPWS decision— identifying its inputs and revealing its internal logic. A lack of supporting information is thought to contribute to delayed responses [78]. Supporting information may include a representation of the GPWS hazard states relative to alert thresholds, or a predictive display of the future flight trajectory superimposed with terrain. However, it is difficult to say how inputs affect an Unstructured decision. Adding inputs to a process potentially makes a decision more informed, but this comes at the cost of attention resources, which is particularly important in time-critical decisions such as collision avoidance. This may negate one of the benefits of alerts: to *hide* complexity by providing an unambiguous, low-resolution signal. In summary, there appears to be a tendency by humans to use Unstructured decision processes in aircraft collision avoidance. While this can be beneficial, Unstructured processes can also be viewed as a departure from the Structured pilot conformance process that is determined prior to operation.

3.4.3 Auto-Overriding Human Control

At the extreme end of the information automation spectrum in Figure 3-19 is "auto-override," which is typically a transient mode in which the human is temporarily removed from the information path. Auto-override is a Structured process that can be viewed as an extension of a decision aid, as illustrated in Figure 3-26. In this model, it is assumed that the same information used by the decision aid is also sufficient for determining the criteria to override human control.

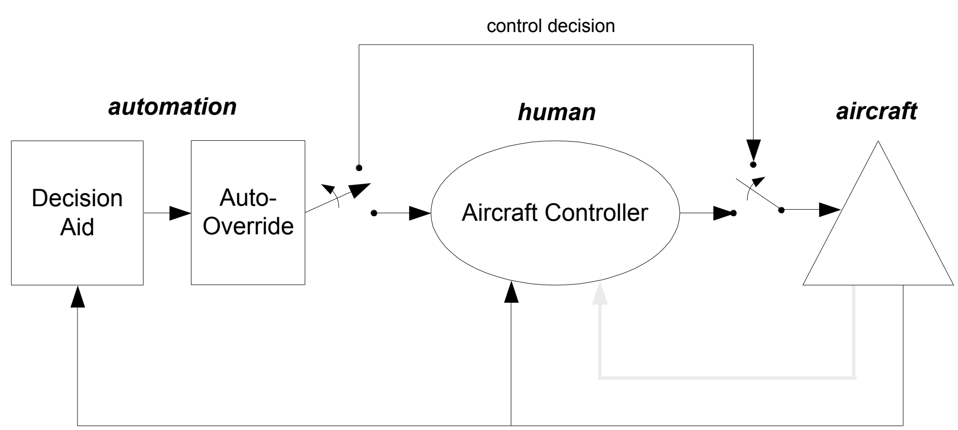


Figure 3-26 Auto-overriding human aircraft control

Viewing auto-override as an automated extension of a decision aid follows naturally from Sheridan and Verplank's scale of "degrees of automation" [138], shown in Table 2. Their spectrum of automation focuses on the functions of decision aiding and controlling, which is a subset of the functions shown earlier in Figure 3-19. Most of the options in Table 2 are true decision-aids, because they keep the human operator in the decision loop. However, designs have emerged which intentionally bypass humans (shown as the 10th item below).

Table 2 Degrees of automation for decision-aid (from Sheridan [138])

1. The computer offers no assistance, human must do it all
2. The computer offers a complete set of action alternatives, and
    3. Narrows the selection down to a few, or
    4. Suggests one, 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 the human, or
    8. Informs him after execution only if he asks, or
    9. Informs him after execution if it, the computer, decides to.
10. The computer decides everything and acts autonomously, ignoring the human.

Decision systems that incorporate an auto-override function are not highly evolved systems. Debates persist over the degree to which automation is involved in a system, and issues on control authority are often the most difficult. However, there is a strong motivation to use auto-override in complex systems, particularly when these systems are intolerant to human errors.

Auto-override in Flight Envelope Protection

While not common, the auto-override function has emerged in some modern aircraft to protect a system from exceeding its safety envelope. For example, the *Airbus A320* uses a fly-by-wire system with a “hard” speed envelope protection and a 2.5g limit. This automation is designed to prevent a pilot from stalling the aircraft, and from exceeding structural limits in an emergency. In the former case, the aircraft directly applies maximum thrust to prevent stall, without the pilot’s consent. Whereas a decision aid would *suggest* that the pilot apply full throttle, “hard” protection systems temporarily disconnect the human. An auto-override function for speed envelope protection is shown in Figure 3-27.

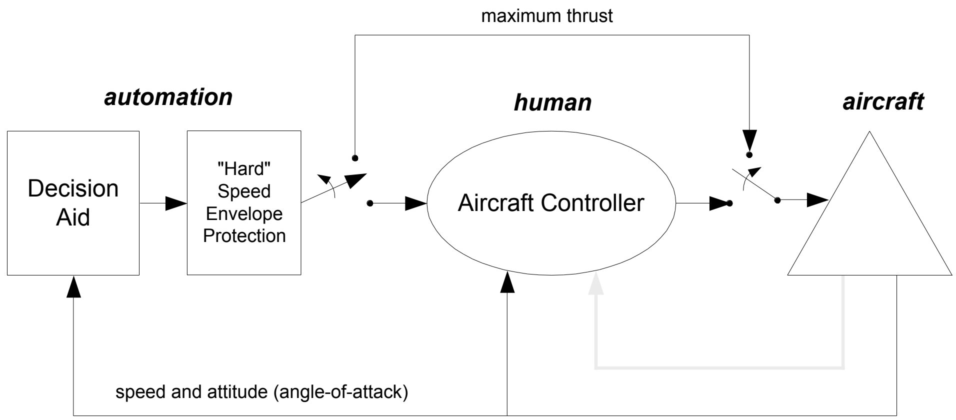


Figure 3-27 "Hard" auto-override for aircraft speed envelope protection

Airbus Industrie officials believe that if the technology exists to automate a function that would prevent a pilot from inadvertently exceeding safety limits, this should be done [61]. This belief inherently assumes that the decision process to override humans is sufficiently understood, and that this information is readily available. The conditions at which aircraft stall occurs certainly fits these criteria, since the aerodynamic characteristics as a function of attitude and speed are well modeled. Of course, it is assumed that these rules apply regardless of the specific context.

In contrast, the traditional view of supervisory control systems has the human in charge at *all* times. As discussed in Section 3.3, there are reasons to believe that the human is better suited to being "in charge."

Each side of the control authority debate has a reasonable argument. For example, on one side, the flight crew is ultimately responsible for flight controls, and responsibility logically requires control authority. Other arguments against auto-override includes issues associated with mode awareness, control transitions in both directions, and physical struggles for control. On the other side, humans also make errors—particularly under time pressure—sometimes with grave consequences. In fact, in nuclear power plants, humans are often *prohibited from intervening* because of potential human error. For flight control, "who" is best may only be objectively determined *a posteriori*, but this is of little help rare events.

"Soft" Auto-Override

Alternate approaches to auto-override represent a different human-automation design philosophy. *Boeing Corporation*, for example, believes that automation is a tool to aid pilots and should not be given authority to override pilot inputs. The *Boeing 777* uses a “soft” approach to speed envelope protection. If the aircraft is decelerating near the minimum speed, automation requires that the pilot apply more force on the yoke through a force actuator. In this case, automation only *informs* pilots of a hazardous situation, similar to an alert, but does so intuitively through the same interface used for control. The critical design feature is that automation informs the human by “fighting” his or her entry of control information.

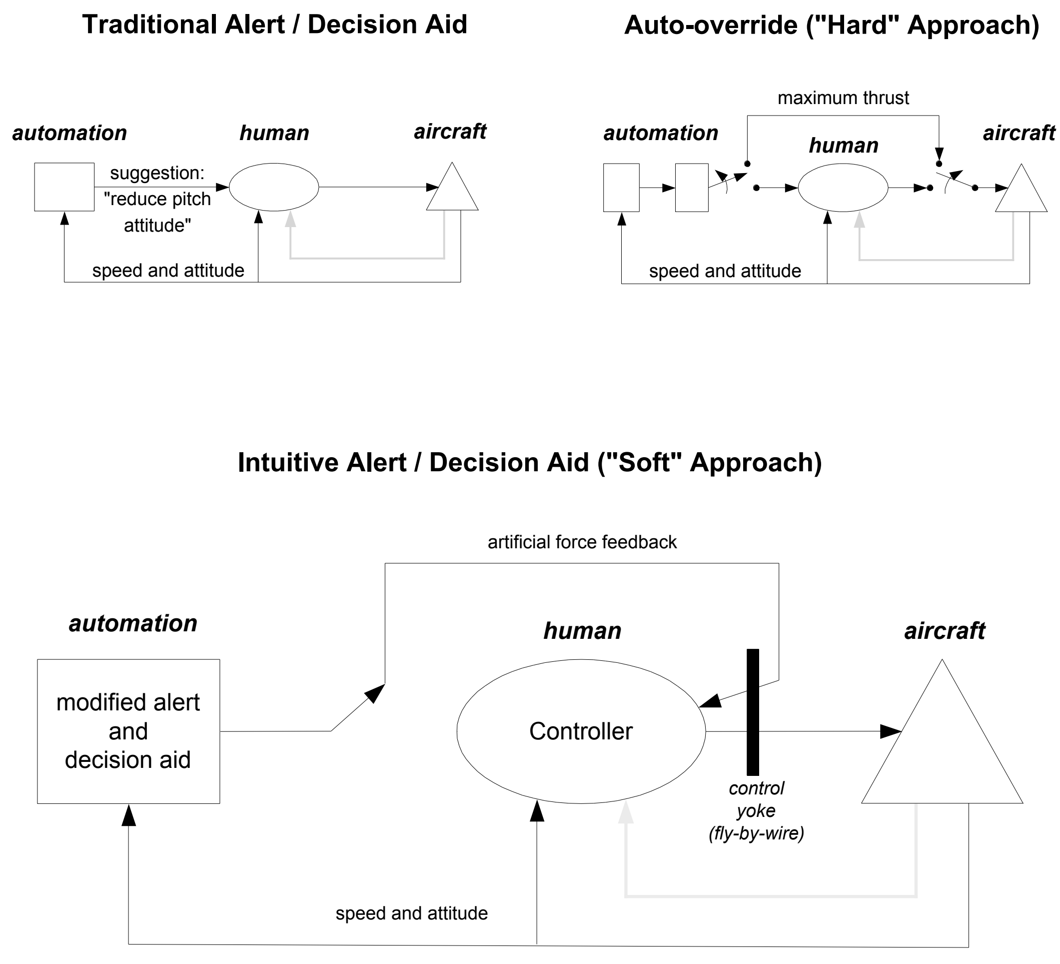


Figure 3-28 "Soft" auto-override for aircraft speed envelope protection

The “soft” approach, shown in Figure 3-28, can be viewed as a compromise between traditional alert and decision aid designs, as in GPWS, and the “hard” approach of auto-overrides. Note that the “soft” approach reduces to the others at each extreme of force feedback: for negligible forces automation acts similarly to traditional displays, while for very large forces automation acts similarly to auto-override mode, preventing humans from entering control inputs.

Both “hard” and “soft” auto-override designs represent different ways of “voting” between conflicting decisions. This issue is further discussed in Section 3.8.3.

3.5 DIAGNOSTICS AND PROCEDURES IN AIRCRAFT AND MEDICINE

This section analyzes standard operating procedures and the information processes that initiate them. A procedure is a Structured decision process that is often executed by humans, but which typically follows a diagnostic process that determines its initiation. Examples are chosen from aircraft (engine fire) and medicine. Each is modeled by a *Structured-Unstructured-Structured* topology, as shown in Figure 3-29. Unlike in supervisory control, here the Unstructured process does not provide high level target states, but instead determines when the conditions are appropriate for the procedure.

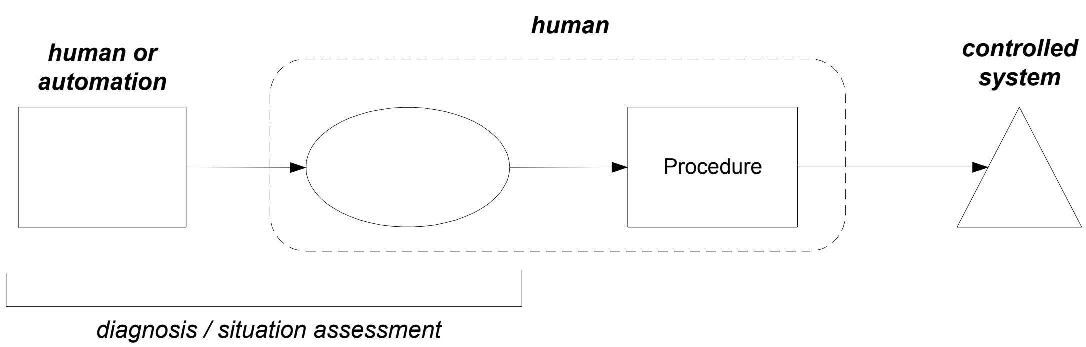


Figure 3-29 Standard operating procedure and its conditional information processes

Since the initiation of a procedure is conditional on an appropriate assessment of a situation, the first two processes in Figure 3-29 provide this function. The information processes prior to procedure execution can include alerts, but *collectively* perform what can be called “diagnosis” or “situation assessment.” While diagnosis is not well-defined, it often benefits from some “front end” processing with rules, such that Structure also appears at the front of the information path.

Despite the ill-defined nature of diagnosis, the result is often a simple, well-defined classification that is used to select and initiate a procedure. The procedure provides the operational decision-maker with a clear set of rules or actions that have been determined prior to operation—sometimes through evolution over considerable time and/or effort [17], [107], [130].

3.5.1 Aircraft Engine Fire Procedure

This section analyzes a procedural decision made during an aircraft emergency—an engine fire—as part of a semi-Structured decision process. Figure 3-30 is a detailed representation of the diagnosis-procedure decision system for an aircraft engine fire. As is typically the case, diagnosis relies on a rich set of information that is processed in an ill-defined manner. In contrast, the procedure requires a simpler set of well-defined information: primarily the diagnosis. Feedback from the aircraft is continually required to monitor the situation as the procedure is executed.

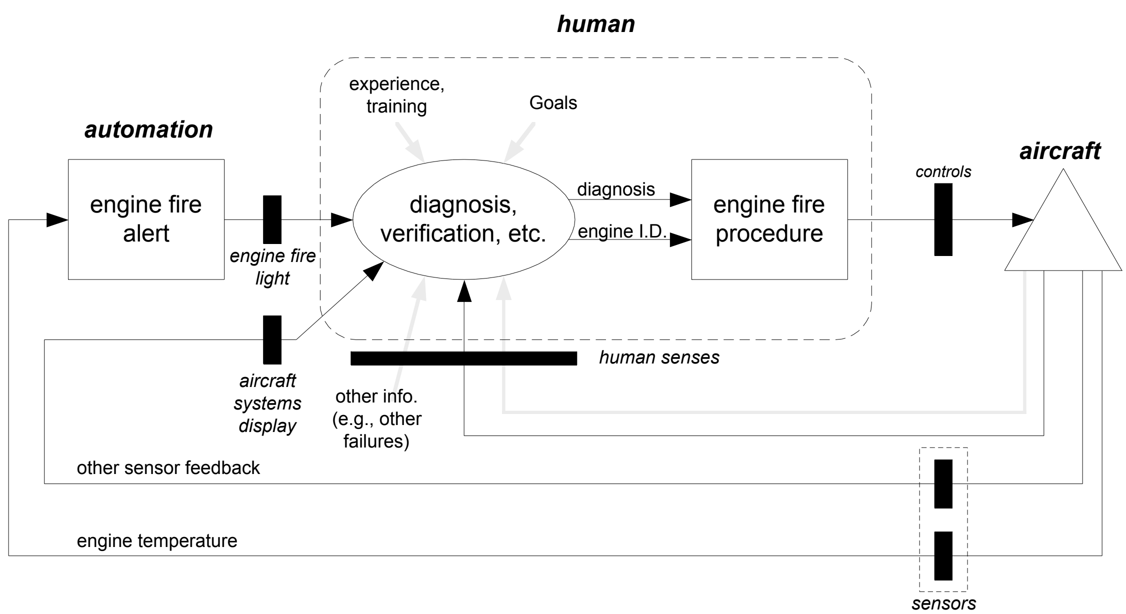


Figure 3-30 Engine fire decision process: *Structured-Unstructured* diagnosis followed by a Structured procedure for resolving hazard

Engine Fire Diagnosis

The diagnosis of an engine fire is a semi-Structured decision process. Pilots are trained to recognize the signs through experience and instruction. While alerts are used to call attention to these hazards via temperature sensors in the engine, alerting logic is based on simple sensing and logic, and one or more members of the flight crew is generally required to verify a fire. Humans can be valuable for incorporating additional information into the diagnostic decision, such as changes in aircraft dynamics, visual and aural information, etc., which are reflected in Figure 3-30 by the inputs to the Unstructured process. In the context of real time information processing, these inputs—including experience—allow humans to accommodate alert errors, potentially improving engine fire diagnosis.

Since the flight crew may not have access to visual information, their contribution to the diagnostic function is not necessarily in improving accuracy. Rather, since they are *responsible* for the safety of the flight, humans require final authority in assessing the situation. This is similar to the arguments for control authority. In this case, the authority is in information processing, rather than in providing corrective actions (the latter is implied by a correct situation assessment). Final authority in situation assessment is reflected in Figure 3-30 by the human as the right-most process prior to the procedure.

Humans also appear to be valuable for assessing a situation in a broad context. Procedures are designed to be used under certain conditions, but these may have to be interpreted during operation based on other "big picture" information. There are occasions, although rare, in which the immediate execution of an engine fire procedure is not appropriate. By incorporating other information and knowledge about the situation in which an engine fire occurs, and understanding higher level goals, a procedure can be applied more appropriately.

Engine Fire Procedure

The procedure for an engine fire is an example of a standardized decision process for resolving a hazard in a timely manner. In the following paragraphs, an engine fire procedure is described. This particular procedure is not universal, since engine fire procedures depend on the specific aircraft model, and, in fact, on the specific airline. The following is from *American Airline*, for a *Boeing 767 aircraft*:

ENGINE FIRE / Damage / Shutdown

Any crewmember noting the engine fire or severe damage shall verbally identify the affected engine. Another crewmember shall verbally verify the affected engine.

AUTOTHROTTLE ………………………………………… DISCONNECT

The pilot-flying will disconnect the autothrottle.

_THROTTLE ……………………………………………. CLOSE

The pilot-flying will retard the throttle to idle.

_FUEL CONTROL SWITCH ……………………………….. CUTOFF

The pilot-not-flying will actuate associated Fuel Control Switch and verify shutdown by checking decay of EGT, N2 and fuel flow.

_ENGINE FIRE HANDLE …………………………………… PULL

The pilot-not-flying will pull Engine Fire Handle.

The above procedure has been pre-determined to be an adequate means for resolving engine fires (assuming it is applied in the appropriate context). It consists of four primitive operations[5], executed by two members of the flight crew. As Figure 3-30 illustrates, the information required for procedure initiation also includes identification of the affected engine. Hence, the procedure operates on a simple set of well-defined information, and results in well-defined actions—both being typical characteristic of Structured processes.

Engine fire procedures provide pilots with a quick way to make effective decisions during an aircraft emergency, and serve as memory aids to supplement training [17], [107]. Given its appropriate initiation, the engine fire procedure can be executed without the need for deliberation or other efforts that require considerable time and attention. Standard operating procedures are also valuable in aircraft decision-making during nominal conditions, providing memory-aiding checklists for aircraft configurations, and providing predictable behavior for air traffic management.

3.5.2 Medical Treatment Procedures

The field of medicine also uses standard operating procedures as part of a semi-Structured decision process. As with engine fire diagnosis, medical procedures are frequently initiated after a semi-Structured diagnosis. However, in this case the diagnostic sub-processes are less redundant: the "alert" provides a clear indication of a problem, but this information is used only to direct a search for a diagnosis via a complex process. Also, human physicians generally execute all decision processes, including the alerting process. A typical medical diagnosis-procedure decision process is shown in Figure 3-31.

Temperature "Alert"

Although patient diagnosis is generally complex, there are often simple indicators that narrow the search space. An example is a temperature "alert," indicates that the patient's body temperature is out of normal range (Figure 3-31). The temperature alert process is not automated, but serves as a well-defined method for calling attention to an important health state.

---

[5] More instructions are provided in case the four steps in the primary procedure are unsuccessful in extinguishing the fire.

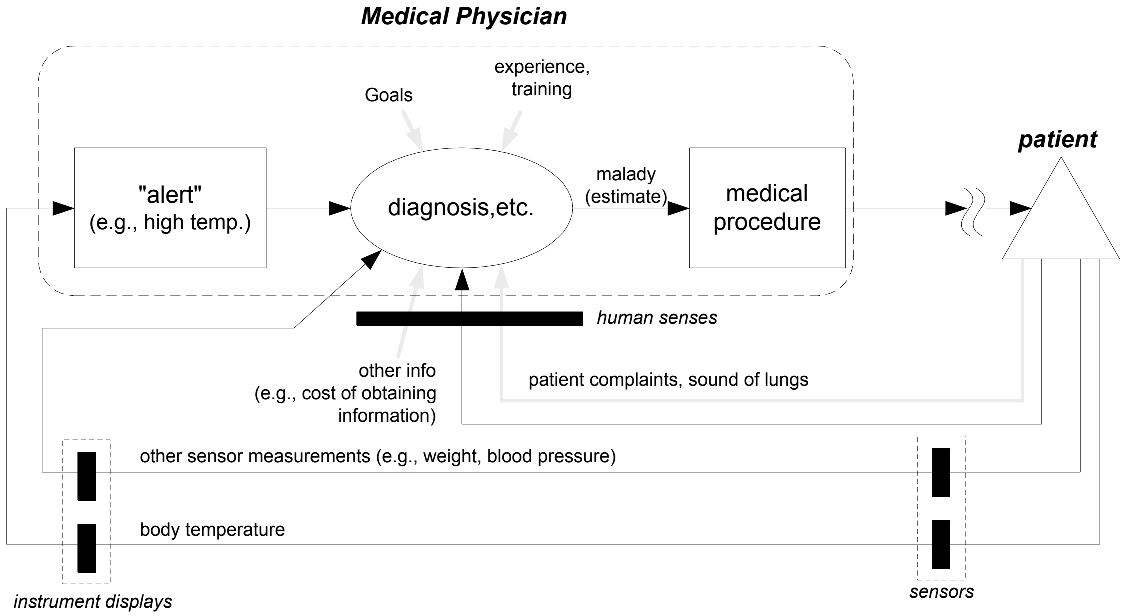


Figure 3-31 Medical decision process involving a standard procedure

A patient typically visits a physician with symptomatic complaints that are verbally communicated—often ambiguously. Due to the complexity of human anatomy, it is important that diagnosis is *efficiently* performed. By classifying body temperature as out of normal range, this coarse information helps the physician determine a likely set of ailments on which to focus.

Diagnosis of Ailment

Following the recognition of a high temperature, a physician considers additional information to support or falsify a working hypothesis of the ailment. The information required for diagnosis is generally complex and ill-defined, including verbal descriptions, visual inspections, and aural patterns such as the sound of lungs during deep breaths. Physicians may also have to deal with inaccurate information that is intentionally provided. Well-defined information other than temperature is also incorporated, such as weight and blood pressure—some of which are obtained with medical instruments. Information for diagnosis is often collected adaptively, based on an exploratory process—although this inner loop is not shown in Figure 3-31 for simplicity.

At some point, a physician judges that a diagnosis is correct with sufficient certainty. This judgment appears to reflect "satisficing" rather than "optimizing" [141], and may consider issues other than diagnosis accuracy. For example, physicians consider the cost of obtaining information and balance this with the risk involved with misdiagnosis and its associated procedures. Diagnosis certainty can always be improved with additional tests, but this is not practical. In short, physicians have to make judgments that consider issues outside of diagnosis accuracy, which involves ambiguity, complexity, insufficient information, uncertainty, and humanistic issues such as subjective judgment, moral judgment, and responsibility. Physicians also understand the associated procedure, and use information other than diagnosis to determine if the situation is appropriate.

Medical Procedure

The resulting diagnosis is often an unambiguous classification of a malady, leading to a well-defined procedure for treating the patient. For example, if a patient is diagnosed with the virus, the physician is not likely to create an *ad hoc* treatment, but will instead use standard treatments such as established doses of prescription medication. Since diagnosis and patient responses have inherent uncertainty, it may be necessary to observe the effects of a procedure (the outer loop in Figure 3-31). Nevertheless, medical procedures provide a prescribed baseline treatment that for treating patients, such that the value of physicians is often in assessing the situation opposed to determining the treatment.

3.5.3 Closing Remarks on Diagnosis

Diagnosis typically leads to decision actions, and can therefore be viewed as the initial part of production rule: the “IF” in “IF…THEN.” In the skill-rule-knowledge classification of human behavior, Rasmussen [120] mentions that in *rule-based behavior*—the use of a sequence of subroutines in a familiar work situation—“the rules can be reported by the person, although the cues releasing a rule may be difficult to describe.” That is, although a procedure can be articulated, its conditional information process—the “IF” or diagnosis—is not always definable.

The Reductionist View of Diagnosis

The diagnosis-procedure paradigm suggests the portion of decision-making that is poorly understood resides at the front of the information path: assessing a situation. That is, the complex and ill-defined portion is associated with the *inputs*, and not the decision outputs. By some means, information is *reduced* to a lower order. This “reductionist” view of decision-making is illustrated in Figure 3-32.

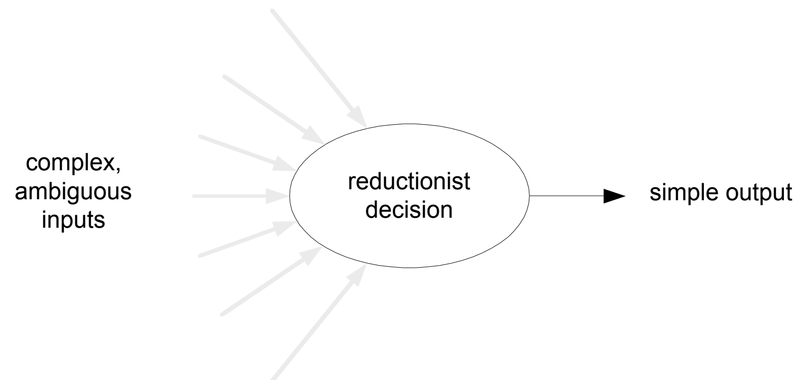


Figure 3-32 The reductionist view of diagnosis

Decision processes are often information-rich, while outputs tend to be simpler and more definable. Consider the prevalence of *binary* decisions: the simplest possible decision output. Example decisions include guilty vs. innocent, take-off vs. hold, etc. These are all simple options, but the decision process may have any level of complexity.

The reductionist paradigm suggests that diagnosis requires an information set that is large in comparison to its output. In familiar situations, humans appear to perform situation assessment perceptually or intuitively, with little difficulty [72]. However, this may change as systems become complex and less familiar.

Diagnosis with Neural Networks

Automated Unstructured processes such as neural networks may be good candidates for ill-defined diagnosis. The reductionist model is, in fact, a good model of actual neuron behavior—biological or computational. A neuron reduces enormous streams of information into a single bit, by either firing or not firing [80]. Although information is destroyed in the process, this approach can be successful only if the destruction is selective—which is what learning/training provides. Reductionism at the individual neuron level often emerges at the network or system level, resulting in a low-order system decision output.

Some of the more successful applications of neural networks are in pattern recognition and failure diagnosis [83], [90], [147]. Both functions are similar in their reductionist nature. Applications continue to grow. Diagnostic systems are being used to monitor factory operations, and are even designed into photocopiers to reduce service calls [156]. In manufacturing plants, visual image processing is often employed using neural networks to perform rapid identification of defective parts. In these examples, the correlation of complex patterns of information to well-defined states is not understood sufficiently to perform with rules. Through supervised learning, however, neural networks can often excel at pattern recognition and diagnosis. Of course, neural networks tend to be successful in domains in which the relevant data is known, and in which training data is available.

As systems such as aircraft become more complex, it is likely that even *experienced* humans will need to rely on automation for diagnostic assistance. This assistance can be from both Structured (e.g., model-based) and Unstructured (e.g., neural network) algorithms, since each offers unique value, based in part on the extent to which a system is explicitly understood. These automated processes may even surpass the abilities of humans. However, if the diagnostic process leads to a well-defined procedure, humans will likely remain valuable during operation for understanding broader issues that are associated with the procedure's execution.

3.6 MULTI-ATTRIBUTE DECISIONS

The selection from a given set of discrete alternatives with multiple attributes is an example of a semi-Structured decision process with a *Structured-Unstructured-Structured* topology, as shown in Figure 3-33. Humans are actively involved in the selection process because formal *a priori* decision models tend to be inadequate for replacing judgment and holistic evaluation of multi-dimensional data [74], [99], [124], [168]. However, since data analysis is both exploratory and complex, the user relies on a set of Structured software tools to manipulate and observe the data with rapid feedback.

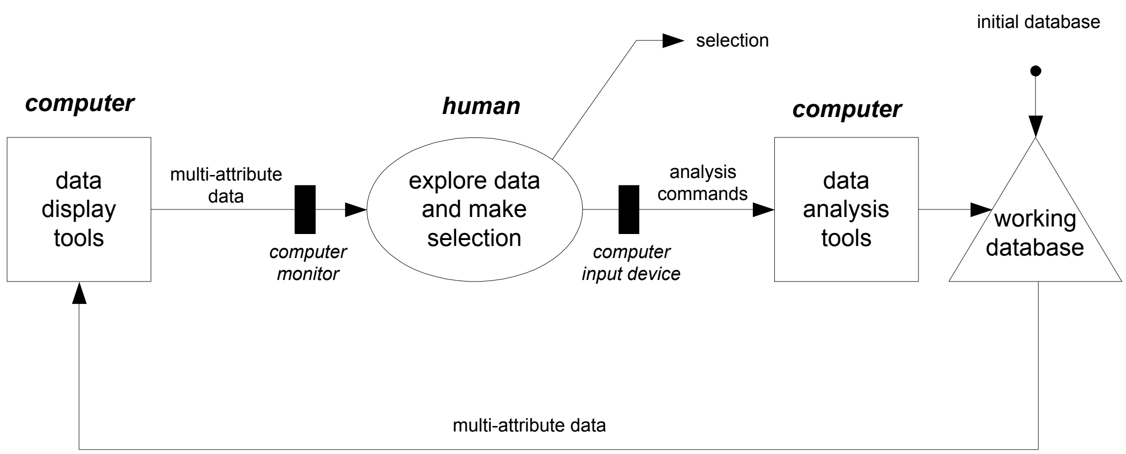


Figure 3-33 Simple model of multi-attribute decision process

The topology shown in Figure 3-33 is similar to that discussed in the previous section on diagnosis-procedure decision-making. However, the application here has characteristics that are more similar to supervisory control and information automation since the human issues *goals* to automation, and the results are observed via automation. An important change from previous examples is that the “controlled system” here is informational: a computer database. One implication of this is that altering the state of the controlled system does not have associated dynamics or environmental disturbances. Furthermore, the purpose of the decision process is not to ultimately alter the controlled system; observing and altering the data serves only as a *means* to the selection of an alternative.

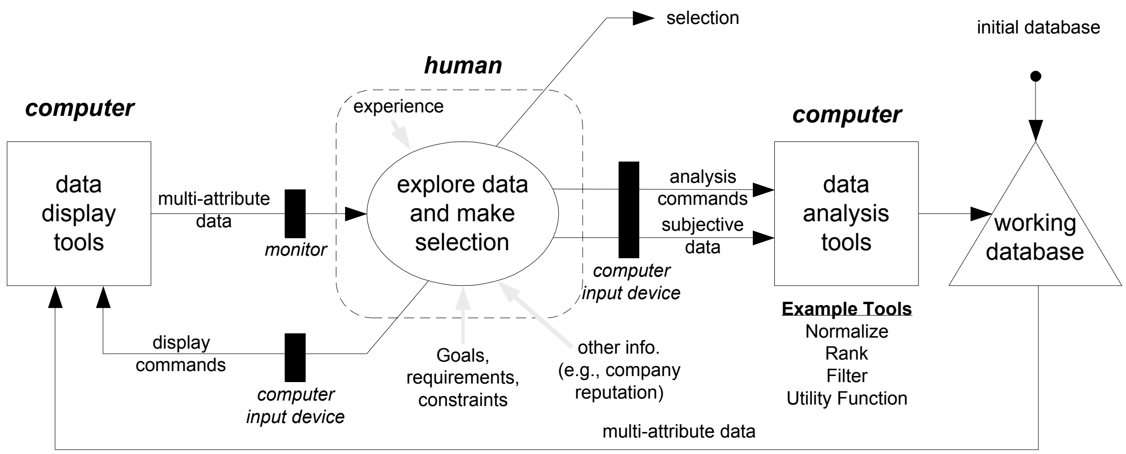


Figure 3-34 Detailed model of multi-attribute decision process

Figure 3-34 shows a more detailed diagram of the multi-attribute decision process. The decision-maker is assumed to enter the situation with a set of Goals, requirements, and constraints regarding the selection. It is unlikely that these can be defined *a priori* to accurately reflect a decision-maker's behavior for all possible data sets. Instead, one must understand the options that are available from a new data set—during "operation"—and make the appropriate trade-offs based on ambiguously defined goals.

Analysis tools such as normalizing, ranking, filtering, and data aggregation (through a decision model) assist by performing calculations that are initiated by simple user commands. Humans also enter additional data, such as subjective weights, preferences, etc. The results of these actions are stored in a working database that is fed back and observed through display logic such as graphics. A separate inner feedback loop allows for display adjustments, which provides an added dimension of data exploration freedom. The exploration cycle can continue indefinitely until an ill-defined stopping criterion is met, and a "most preferred" (versus optimum) selection is made [74].

3.6.1 Data Assumptions

The "database" in Figure 3-34 is assumed to contain information for decision alternatives D_n (n = 1 to N), each with associated attributes A_m (m = 1 to M). As Figure 3-35 shows, the data can be represented as an N x M (*5 x 6*) matrix, which is the typical form for spreadsheets. The function of the decision system is to ultimately select a single *row* from this matrix.

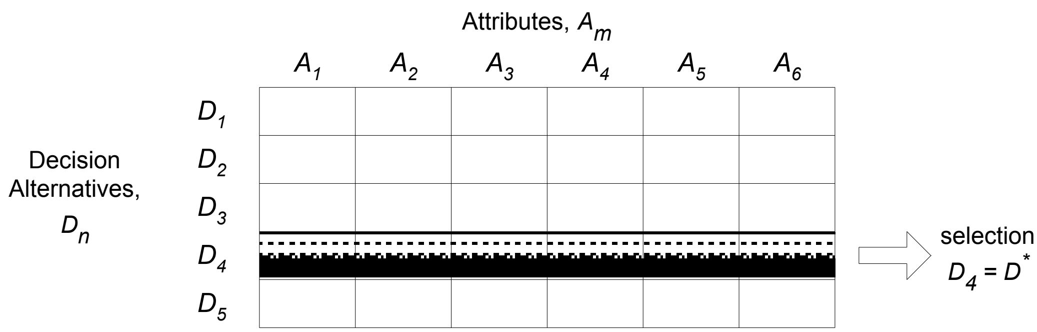


Figure 3-35 Multi-attribute data represented as a matrix or spreadsheet

For this analysis, it is also assumed that the data for the matrix is numerical. This allows all data to be transformed through mathematical and logical operations. Qualitative information is assumed to come from other sources. For example, in selecting an electric motor (the example used throughout this section) a catalog database may contain numerical attributes such as cost, weight, speed, torque, and operating voltage, and not qualitative information such as pictures. Although multi-attribute data is not limited to products from a catalog, this will be the example used in the remainder of this section.

3.6.2 The Problem of Order Reduction

The multi-attribute selection problem is one that is fundamental to decision-making: how to integrate multiple pieces of information into a single decision. It is a problem of order reduction, (as discussed in Section 3.5.3). Since each alternative D_n is a *vector* of attributes, in which each *element* is an attribute value, alternatives seemingly have to be compared in multi-dimensional space, and their order reduced through techniques such as elimination and aggregation (Section 2.7.1). The issue of reductionism is certainly not unique to multi-attribute decision-making, but is apparent here since the information components are elements a vector.

The fundamental problem with multi-attribute decision-making is determining how to operate on individual attributes—the “parts” of the decision alternative—without sacrificing the "wholeness" of the decision task [85].

3.6.3 The Value of Humans in Multi-attribute Decisions

Humans are valuable as a functional component of a multi-attribute decision for many reasons. In particular, it appears that they can incorporate multi-dimensional data *holistically*—without explicitly considering how each attribute individually affects the overall value of a decision alternative. This is believed to be a product of experience and familiarity, as in the pattern recognizing abilities of experts. While holistic decision-making is not always possible (which will be discussed shortly), it is valuable because it does not force the human to follow a deliberative, analytical process, which has been shown to sometimes reduce the quality of decisions [36], [53], [72].

Additional Information

Humans are also able to incorporate other information into the decision, beyond the attributes that are available from the database. For example, it may be important to consider a company's reputation on delivery time and technical support, or information about aesthetics—qualities that are not likely to be found in a catalog database [62]. Since any decision model is based on a limited set of information, it may be important to perform "sanity checks" on its choice by incorporating additional information.

Adapting to Data

The ability to adapt to the data is also an important characteristic of human decision-makers. Adaptation can be required in different ways, such as the following:

- *Missing data* – When elements or columns of data are missing from a matrix, a selection may need to be made despite that the decision-maker is under-informed.
- *Additional data* – When additional attributes are available in a catalog, it may be beneficial to incorporate this into a decision.
- *Tightening goals and constraints* – Given a set of data for which there are a greater selection of options than expected, one may want to consider adjusting goals and constraints to further improve the quality of the selection.
- *Relaxing goals and constraints* – Given a set of data for which there are few or no alternatives that meet an initial set of criteria, it may be necessary to adjust the goals and constraints so that a reasonable selection can still be made.

Whereas humans can often accommodate the above situations, the rules underlying such adaptive behavior are not well understood [134]. These are some of the reasons why *a priori* decision models are inappropriate for multi-attribute decisions.

3.6.4 Multi-attribute Decision Tools

Some of the techniques that have evolved in multi-attribute decision-making (also called *multi-criteria* decision-making) are labeled in Figure 3-34, such as ranking and functions. In the broader scope of decision support systems, which according to Simon [145] assist in four phases of a decision process (*intelligence*, *design*, *choice*, and *review*), these tools address the *choice* phase [115]. Within this phase, decision tools can be viewed as assisting the decision-maker through one of the following *general* functions:

- Reducing complexity – As mentioned, the multi-attribute decision problem is one of *order reduction*. When the size of a matrix is large, cognitive limitations in short term memory and information processing may emerge. The size of the matrix is reduced in many ways, such as through graphical operations [47], or more directly by deleting rows (alternatives) and/or columns (attributes). In addition, scalar utility functions also reduce complexity through aggregation.

- Changing the representation – The manner in which data is represented can help people "see" different aspects of the data. Insight can be improved through representations that exploit perceptual pattern recognition, experience, etc. Representational tools are not only for visual display, but also for altering numerical data (e.g., normalization).

The following describes some of the highly evolved tools that perform the above general functions.

Filtering

"Filtering" is a term used to describe the adding or deleting of alternatives based on well-defined criteria. The size of the data can be reduced directly by deleting rows (decision alternatives) or columns (attributes) of the matrix. The elimination of *attributes* is generally a manual task, since one has to judge which attributes are not relevant or important for the task.

However, the elimination of *decision alternatives* can be accomplished by specifying acceptable numerical ranges on attributes, which can be automatically applied to all rows in the matrix.

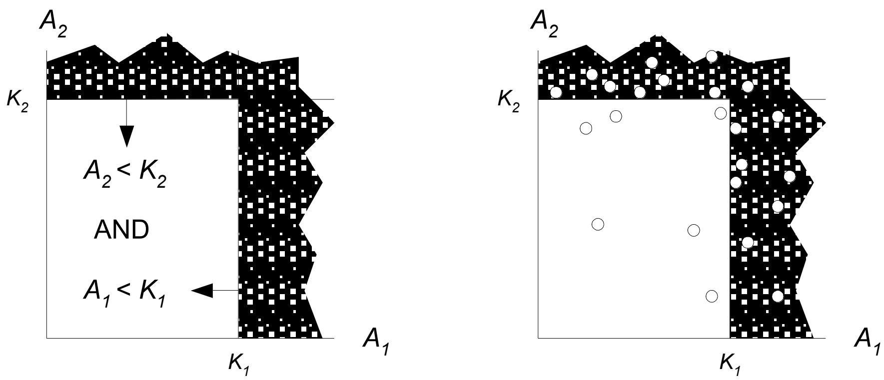


Figure 3-36 Attribute criteria can be used to "filter" decision alternatives by defining an acceptable region (unshaded)

Figure 3-36 graphically illustrates how filtering can be used to reduce the number of alternatives from 23 to 6 through a logical ("AND") combination of individual attribute thresholds K_1 and K_2. The decision-maker can then focus only on the small number of "acceptable" alternatives. However, in Figure 3-36 the acceptable region is constrained to be rectangular, which may not reflect actual choice behavior. The decision-maker may desire to adjust these constraints to better adapt these simple filtering rules to the data set and the ill-defined selection goals.

Ranking

Ranking or ordering is fundamentally a reductionist action because it represents a single dimension of evaluation. This is true whether the ordering is cardinal (with a numerical scale) or ordinal [27].

Unless ranking is done subjectively, it generally requires a numerical order [167]. Rankings that are based on a *single* attribute obviously do not reflect the values of other attributes. At the other extreme, rankings that are based on utility functions (which may incorporate all attributes in the data set) may not reflect the complex, multi-dimensional considerations that a decision-maker may have. In addition, the single dimension of ranks cannot reflect the intransitive behavior that people often demonstrate [75]. Nevertheless, the order-reducing characteristics of ranks provide humans with an estimate of good and bad alternatives, whose position in the order can be easy to track during the exploration process.

Multi-attribute Utility Functions

Utility functions are analytical *models*, which are a special use of Structured processes. With models, not only are the inputs and outputs symbolic representations, but the process itself is a representation—in this case a representation of a decision-maker's value or "utility" of a decision alternative [27].

As with ranks, utility functions are beneficial because they provide a representation of the overall (scalar) value of an alternative, which helps a decision-maker separate good and bad alternatives. Utilities for individual or coupled attributes are elicited from the decision-maker prior to decision-making. A multi-attribute utility function is typically an expression that aggregates subjectively "weighted" utilities. These weights abstractly reflect the relative importance of attributes. Figure 3-37 shows a simple additive multi-attribute utility function, in which w_i are the weights, and $U(A_i)$ are the individual utilities—both of which are elicited prior to decision-making.

$$\text{attribute data} \longrightarrow \boxed{U(\boldsymbol{A}) = w_1U(A_1) + w_2U(A_2) + w_3U(A_3) + \ldots} \longrightarrow \text{utility}$$

Figure 3-37 Example of a multi-attribute utility function

Utility functions provide an approximation to actual (holistic) value, which can be valuable even in the simplest models [27]. However, they are often insufficient for reflecting actual choice behavior, particularly when the model is explicitly articulated prior to operation [38], [18], [60], [74], [111].

Normalization

Normalization is an example of a more general action that changes the representation of data, and does not necessarily reduce its order. Typical normalizing actions are dividing a number by a reference (e.g., transform to a percentage) or subtracting off a reference to yield a difference. For example, in selecting an electric motor, one may choose to normalize all values of *stall torque* by the value corresponding to the motor that is most familiar. The "best" representation, if it exists, depends on the user and the task. More generally, each representation offers different insight.

Graphics

As discussed in Section 3.4.1, graphical transformations provide a representation that allows humans to exploit their natural perceptual abilities. When users interact with graphical tools—through the inner loop in Figure 3-34—it is not necessary to make explicit what is being sought. Rather, analysts can make discoveries by adjusting well-defined display parameters and observing ill-defined patterns.

In multi-attribute decision-making, graphics are a powerful tool particularly for navigating large data sets. The same data presented numerically can be more difficult to observe due to limitations in symbolic processing [141] and short-term memory [93]. While humans tend to be good at recognizing data trends, patterns, clusters, and outliers, they are limited to viewing data in low dimensions. Various graphical transformations have been developed to help navigate through high-dimensional space [41], [47], [48], [157]. Many tools have evolved which combine the Unstructured perceptual pattern recognition abilities of humans with Structured graphical transformations for displaying revealing views of low-dimensional data.

One manner for displaying high-dimensional data is through "scatterplot matrices." In this technique, three-dimensional data (which here is considered "high" for simplicity) is displayed as a matrix of two-dimensional subplots, in which a data point represents two attributes of a three-attribute decision alternative. Although the scatterplot matrix does not hide any data, only two-dimensional patterns can be perceptually recognized. It becomes more difficult to "see" three-dimensional patterns because this requires a non-intuitive correlation between subplots.

Regardless of the data size, the graphical representation of data can make patterns and relationships salient, and more fitting to the user and task. The rules graphics are always determined *a priori*, to a certain extent. However, it may be beneficial to add more flexibility by allowing the user to adjust parameters that are normally fixed prior to operation.

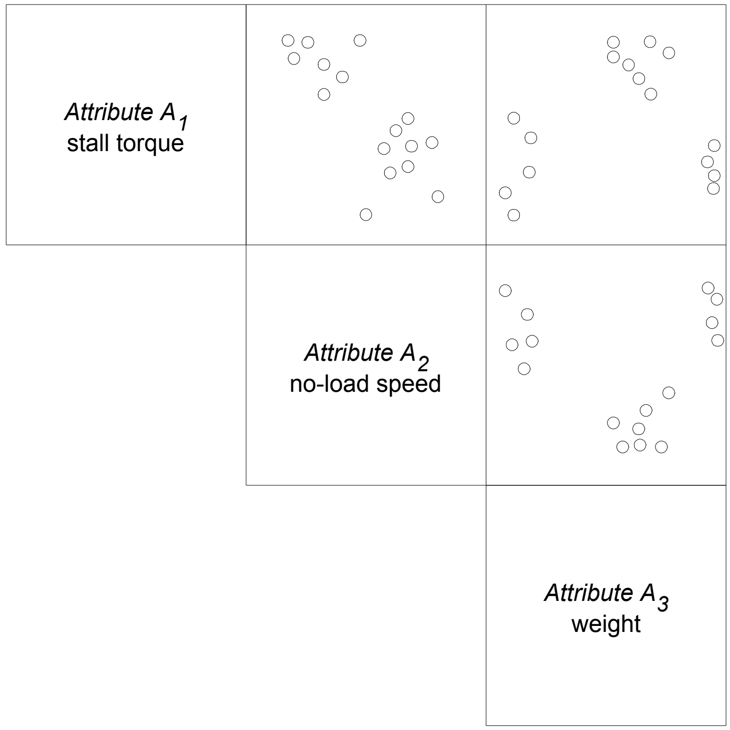


Figure 3-38 A scatterplot matrix is one graphical tool for observing high dimensional data

3.6.5 Highly Interactive Tools

The tools for data analysis and display offer an added degree of value when they are "highly interactive." While any of the previous tools can be considered interactive based on the feedback loop in Figure 3-34, highly interactive tools allow the user to modify parameters that are traditionally determined prior to analysis. This adds Unstructure to the operational decision process. In addition, highly interactive tools exploit the computer's ability to perform rapid calculations for cause-effect exploration, which may allow another type of learning due to the perception of *dynamic* patterns.

Since multi-attribute selection is an exploratory task with ill-defined goals, the Unstructured decision process cannot have associated rules that are defined to be *a priori* optimal for the system. However, interactive rules can be better matched to the data, the task, and the user, allowing the "best" rules to be determined during operation through an adaptive learning process.

The tools described in the previous section can each be designed to be highly interactive. This section discusses two of these tools in this respect: interactive *utility functions*, and interactive *graphics*.

Highly Interactive Utility Functions

Earlier it was mentioned that humans can find it difficult to select subjective weights in a utility function to accurately reflect their actual choice behavior. Although all models are an abstraction, and thus prone to errors, the "errors" in utility functions cannot be measured in a formal sense when the goals are ill-defined. However, the user can compare the results of the decision model (utility function) with evaluation during operation. By observing the results of parameter changes (e.g., weight adjustments), utility functions can evolve during operation to reflect actual choice behavior more accurately than with static models [6], [11], [13], [74], [99], [148]. Also, by making parameters such as weights explicit, people are forced to clearly articulate information about goals and constraints that previously may have been vague.

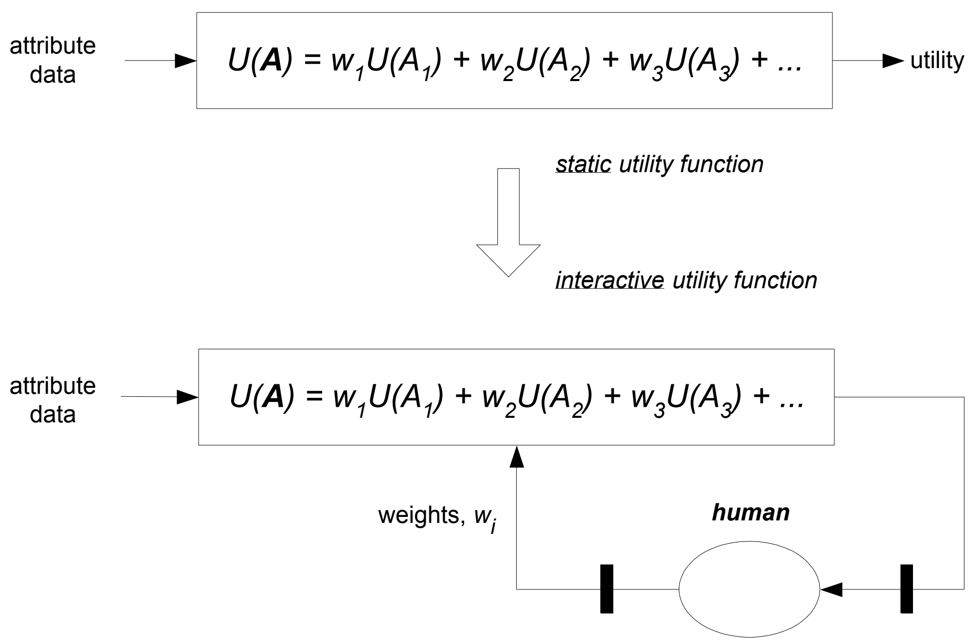


Figure 3-39 Static and interactive utility functions

In short, deferring the *complete* determination of the utility function until operation (e.g., through interactive weights) appears to add value to the multi-attribute decision process. Interactive utility functions may improve multi-attribute decisions by allowing the human to refine the model during operation. This not only can improve the accuracy of the model for a particular data set, but it also forces decision-makers to articulate model parameters such that their preferences are understood more explicitly.

Highly Interactive Graphics

Highly interactive graphics allow the user to adjust parameters whose value may have little significance in a static sense. While traditional graphical tools provide representations that have static value (e.g., pie charts), highly interactive graphical tools add features to perceive data patterns through high-frequency interaction [89]. For this reason, "highly interactive" graphics is sometimes used interchangeably with "dynamic graphics."

Highly interactive graphics can be especially powerful for exploring high-dimensional data (greater than two or three dimensions). An example of a semi-Structured approach for navigating high-dimensional data is through "brushing"[23]. The same scatterplot display of the attributes of an electric motor from Figure 3-38 is shown again in Figure 3-40. However, in order to understand data relationships across these subplots, a technique called "brushing" is employed. In brushing, the same decision alternatives corresponding to the selected points in one subplot are visually identified in the other subplots (e.g., through shading or coloring). As the brush is moved—perhaps identifying clusters in one subplot—high-dimensional patterns may become apparent by observing its effects in the other two subplots. Other highly interactive graphical techniques, such as those based on animation, are discussed in [23], [34], [76].

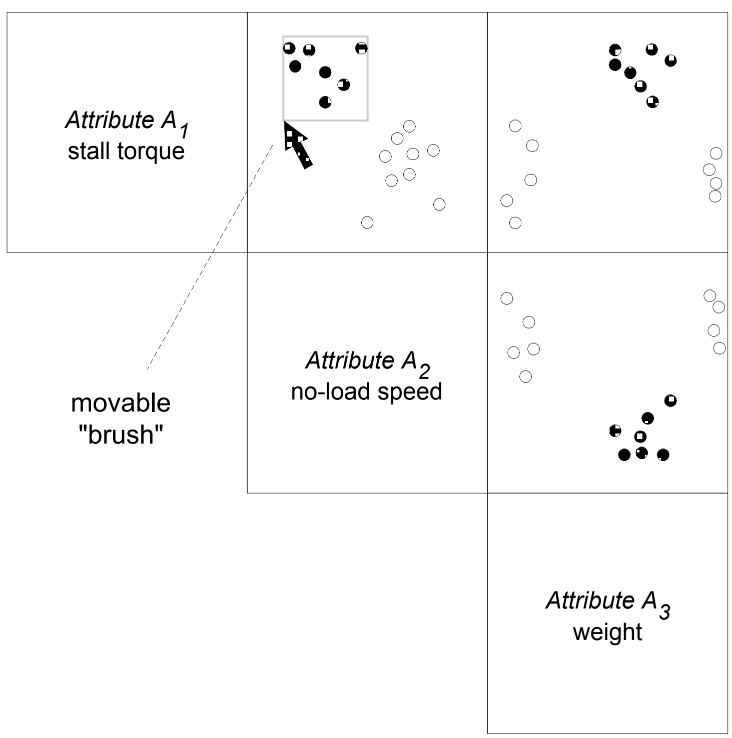


Figure 3-40 "Brushed" scatterplot matrix: an example of highly interactive graphics

Interactive graphical transformations such as brushed scatterplot matrices also illustrate why multi-attribute decision-making has an ill-defined stopping criterion for exploration. For Unstructured decisions, the concept of satisficing may apply, which states that a decision-maker does not optimize, but seeks a “good enough” alternative with a reasonable amount of effort. The efforts involved in exploration are a “cost” compared to the “benefits” of understanding data, yet neither cost nor benefit is clearly definable. At some point during analysis—perhaps when a perceptual pattern of data is discovered—the user is “satisficed,” and judges that further analysis is not required. In multi-attribute decisions, this is the point when a selection is made.

3.7 ENGINEERING DESIGN

3.7.1 Introduction

The engineering design of reasonably complex artifacts can be modeled as a semi-Structured decision process with an *Unstructured-Structured-Unstructured* topology, as shown in Figure 3-41. When a product is complex, design occurs in phases, beginning with conceptualization. After this creative, ill-defined process, design concepts are analyzed using formal models and analytical techniques—typically with CAD (computer aided design) tools. Finally, since design problems are typically open-ended, evaluation of a concept requires human judgment. The three processes in Figure 3-41 are common in most engineering design activities [146].

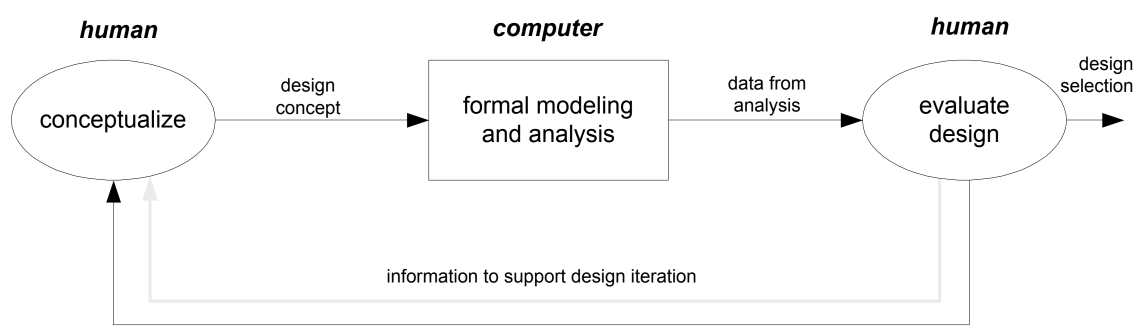


Figure 3-41 Simple model of engineering design process

Engineering design differs from industrial design in part because of the formal analysis that is involved. “In the middle” of the design process, there are parts that are well understood to the extent that they can be prescribed with rules. For example, a Structured process can be used to predict the deflection of a steel beam through engineering models. The calculations were historically executed by human analysts, but today are executed by computers. In order to do so, humans must generate designs and evaluate design concepts through a human-computer interface.

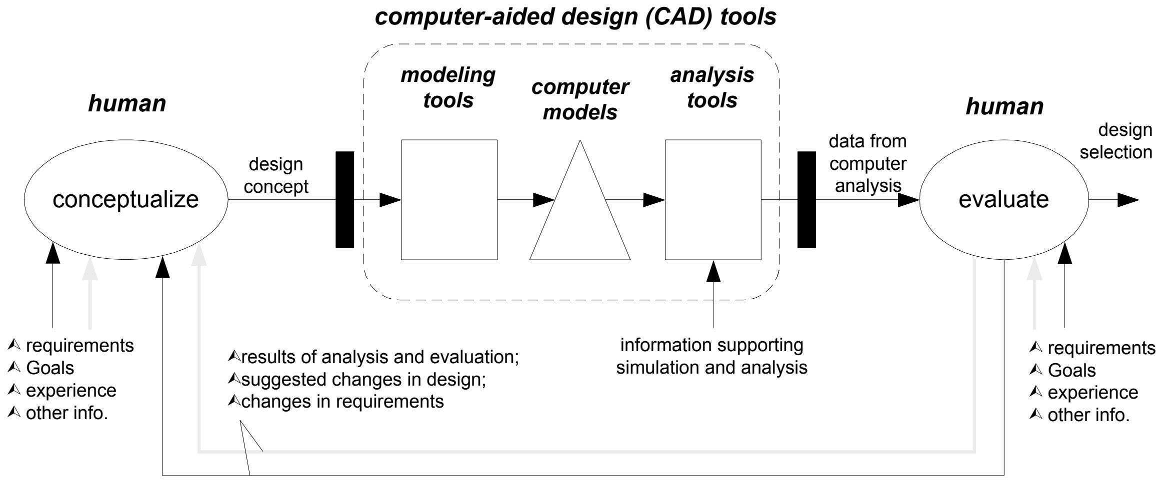


Figure 3-42 Detailed model of engineering design process

Figure 3-42 illustrates the design process in more detail. Engineering design begins with a definition of requirements, which are then translated into initial concepts using ill-defined processes. The conceptualization process not only requires creativity and other humanistic qualities, but its output must have a representation that is appropriate for formal analysis. Modern engineering design is therefore *constrained* by the ability to generate this representation. In fact, many of the CAD tools are dedicated to assisting humans in building the model, such as a geometric representation of an engine component. Once this is done, the computer model is analyzed by applying simulated scenarios such as environmental loads, manufacturing constraints—often with separate tools. The results of this analysis are then evaluated with respect to formal requirements (e.g., manufacturing cost, weight) and informal human Goals (e.g., aesthetics, ergonomics, moral judgment), eventually resulting in a design selection.

The *Unstructured-Structured-Unstructured* paradigm for design is a model that illustrates the semi-Structured components at one level of abstraction. In practice, design can involve many sub-loops and hierarchies within each of the phases. For example, designers may iterate on a computer model before any formal analysis is performed, or may progress in more detail during each iteration to reduce cost in early cycles. Also, some Structure can often be useful within the conceptualization and evaluation phases. Nevertheless, the semi-Structured process in Figure 3-42 illustrates where the majority of Structure has evolved within human-computer systems for engineering design.

3.7.2 Translating "Whats" to "Hows"

Design starts with a definition of the functional requirements of a desired product. As mentioned in 1.3.4, these are a description of "what" is needed—the design task is ultimately to produce an unambiguous description of "how" to achieve these goals [86], [146]. In the case of engineering product design, the "how" is ultimately a detailed representation of the product (e.g., geometry, materials) and often broader information associated with its manufacture, assembly, operation, recycling, etc. Design can therefore be considered a process whose primary inputs are functional requirements (the "whats"), and whose outputs are a set of primitive, unambiguous representations (the "hows"). This is illustrated in Figure 3-43, in which the type of process is not distinguished.

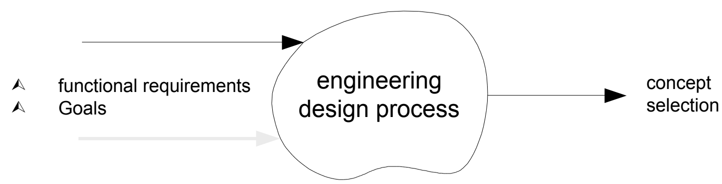


Figure 3-43 General input-output model of the engineering design process

The input-output mapping is, in some sense, no different than any other processes described in this chapter. Simon [141] uses the term "design" to refer to a broad class of non-trivial decision-making, and believes that there is nothing inherently humanistic about the process: its complexity is a merely a consequence of a very large search space. Nevertheless, it is evident that the decision processes that humans use to make the leap from high-level requirements to detailed design concepts are poorly understood.

Design Goals

One of the reasons why design cannot be sufficiently accomplished with only Structured processes is because of the characteristics of the design goals [21]. While the functional requirements of engineering products typically *include* explicit, objective performance metrics, these alone are inadequate in part because:

- Some requirements are *ambiguous* – Attributes related to "customer satisfaction" and aesthetics can be critical in design, but are difficult to explicitly represent.

- Some requirements are *uncertain* – Goals may be prescribed as "hard" constraints early in the design process, but these may change unpredictably during the design process as more information is obtained from initial concept analysis, marketing, etc.
- Requirements are *insufficient* – Any finite set of requirements is a simplified abstraction of an actual set, and it is the designer's responsibility to make this abstraction. However, in engineering design this can be particularly problematic due to the breadth of issues that are involved. Many of the design goals are implied, but never articulated [42].

Human designers are observed to accommodate these characteristics of design goals. They have an inherent understanding of *implicit* goals, which can supplement any articulated set of design requirements. This includes qualitative goals such as aesthetics and style. Perhaps the most valued characteristic of humans is their ability to not only understand design requirements, but to translate these inputs into reasonable designs concepts.

3.7.3 Design Conceptualization

The first phase within the engineering design process in Figure 3-42 is conceptualization. This is the portion most often associated with the term "design," as it typically requires creativity—both in the sense of finding novel solutions within a very large search space, and also in the aesthetic sense (Section 2.7.10). The lack of understanding associated with *creative* decision-making is perhaps the primary reason why the generation of good design concepts is difficult to accomplish with rules.

Structured Conceptualization

The generation of creative concepts with rules is difficult in part because of the ambiguity and large size of the search space. Ambiguity originates with the design goals, which involve qualitative and subjective descriptions. When the goals cannot be made explicit, the basis for generating concepts is ill-defined. However, even when design goals can be made explicit, it may still be difficult to search the parameter space with reasonable efforts, since random trial-and-error is impractical [96], [141].

The Value of Humans

Humans appear to generate good design concepts using a combination of top-down and bottom-up reasoning. By understanding functional requirements top-down, they can potentially generate novel designs without biases from previous designs. Yet they also use experience to generate concepts bottom-up—considering practical constraints such as materials and manufacturing. Although experience generally imposes design biases, it is valuable for providing the necessary heuristics in order to search for *reasonable* concepts.

One of the most valued characteristics of humans is their ability to generate design concepts that involve aesthetic and other subjective judgments. Since humans have a good sense of *implicit* design goals, they are not constrained by what can be explicitly represented—one does not have to know precisely what is being sought in order to "find" it. Humans appear to rely strongly on nonverbal thinking and intuition for generating creative designs [42].

CAD systems have not replaced the human conceptualization of engineering designs, but have enhanced many aspects of the process through modeling tools [45]. The most common example is solid geometric modeling, which helps provide an intuitive, visual representation of a design concept. These may enhance creativity by providing a quick means of translating a vague idea into a concrete model that can then be explored in detail. The computer can also be viewed as a creativity constraint, limiting representations to surfaces and shapes that can be generated through CAD modeling tools. However, this representation constraint also allows the computer to be exploited for formal analysis.

3.7.4 Computer Modeling and Analysis

Given an appropriate model of a design, the digital computer provides a powerful medium for analysis. For domains that are understood sufficiently well (which may be the case for only a subset of the relevant domains), explicit models can generate valuable data that may be difficult to understand intuitively. For example, computer models can quickly calculate the oscillatory behavior of a complex steel frame—a possible but tedious task for humans. A variety of CAD analysis tools exist for different domains, each of which may require a model specific to the tool.

Computer models allow a designer to understand more information associated with a particular concept. They not only provide a way to generate the consequences of uncertainty in design parameters, but also provide a way to easily explore parameter space through "what-if" analysis. In either case, models do not *determine* good or bad designs, but generate more information that is useful for evaluating a design. Models can be viewed as a type of "observer" because they transform one set of data (e.g., external loads) into another set (e.g., internal stress) that does not represent a decision output, but has added value to a decision-maker. Computer models are a special application of rules that are intended to make design evaluation more informed.

Of course, every model is an abstraction, and therefore has errors. Rouse [128] provides a categorization of these errors and their impact in decision-making. Modeling errors should be taken into account during evaluation, even when errors cannot be characterized, in which case human judgment may be required to use the results of computer models appropriately.

3.7.5 Design Concept Evaluation

The evaluation of a design concept involves both formal and informal techniques. Design specifications or functional requirements are often in the form of explicit performance objectives that can be easily checked with formal rules—such as if a steel beam meets the load requirements. However, there are many other attributes of a design that are qualitative, open-ended, and whose *system* implications are not definable through trade-offs. In addition, engineering design often requires humanistic considerations such as moral judgment and responsibility.

Qualitative Attributes

Qualitative attributes such as aesthetics and style can be important factors in engineering design. These attributes may be in conflict with performance-oriented attributes, but any design must consider a product as a *system*. Although humans cannot articulate what is required aesthetically from a design concept, they can know beauty when they see it, for example, and are therefore necessary in the evaluation process.

Information Integration

Since engineering design involves so many considerations—performance, cost, ergonomics, manufacturing, assembly, aesthetics, etc.—the evaluation of a concept involves the *integration* of information from many different domains (Figure 3-44). While CAD tools for modeling and analysis tend to be domain-specific, the information they produce has to ultimately be integrated at the system level. Humans appear to do this without defining explicit trade-offs. Furthermore, humans can use other information that is not directly available from formal analysis, and can incorporate a lifetime of experience into their judgment. Just as design goals are broad and complex, the evaluation of a design with respect to these goals requires the integration of many design attributes in a way that is poorly understood.

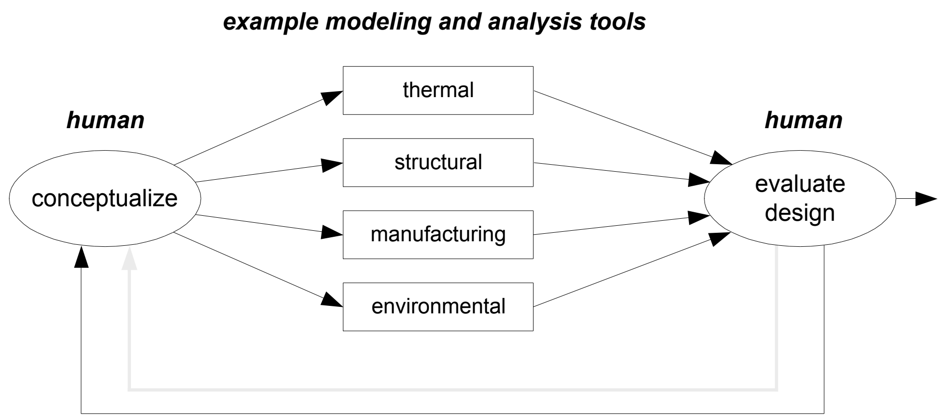


Figure 3-44 Evaluation of a design requires integrating data from multiple domains (right)

Moral Judgment and Responsibility

Engineering design (unlike industrial design) often involves ethical issues, such that moral judgment is needed in evaluation. For example, engineers have an obligation to society for choosing safe designs, and need to judge when a concept is "safe enough"—since added safety is in often conflict with other design goals such cost or performance. Also, when people are legally responsible for a concept selection, there is an incentive to ensure the design is satisfactory.

Feedback in the Design Cycle

Humans can be important in the design *cycle* for understanding what is needed in order to support the efficient generation of new concepts. Part of evaluation is not only determining *if* a concept is insufficient, but also determining *what* types of changes are required.

When evaluation and conceptualization are performed by different people, information about design iteration (the feedback in Figure 3-42) has to be explicitly communicated. This information may not only include the results of formal evaluation (e.g., percentage short of an objective performance requirement), but also less formal information such as subjective judgments of why a design concept is inadequate. In fact, design evaluation may bring the functional requirements themselves into question, in which case the next design iteration will have a modified set of goals. The closed-loop design process, while similar in a sense to feedback control, is too complex to simply measure or even define an "error" between a desired and modeled design concept. Ill-defined design goals—which are inherent to any open-ended design problem—require not only an ill-defined evaluation, but also require an ill-defined process for determining the type of information required to support the next design iteration.

3.7.6 Comparison of Design to Multi-attribute Decision-making

This section makes comparisons between engineering design and multi-attribute decision-making in order to recognize the factors that contribute to Unstructured processes. Multi-attribute decisions and engineering design decisions share the following characteristics:

- **Both are *exploratory*** - Design is iterative in part because of its exploratory nature. Many of the implications of design decisions are not initially understood—one cannot "see" so far ahead in the process to immediately understand the effects of a design concept. CAD tools enhance a user's exploration of designs by quickly generating results from engineering models. Similarly, multi-attribute decision tools are used to understand *existing* data. In both cases, the evaluation of a set of data and the modifications to the Structured tools are based on an ill-defined decision process. Also, exploration is feasible because the "controlled system" (the triangle in the diagrams) is *informational*, not physical.
- **Both involve *adapting*** - The initial goals and requirements may be defined without a sufficient understanding of their ability to be fulfilled. As information is obtained during exploration, the goals may have to adapt to accommodate this information. For example, in multi-attribute decision-making, the initial constraints may need to be relaxed in order to yield a reasonable set of alternatives.
- **Both involve *complex data integration*** - While CAD tools generate data from multiple domains (e.g., thermal, manufacturing, environmental), humans are required to integrate these often-incommensurate "pieces" as part of evaluation. This requirement was also evident in multi-attribute decision-making, in which each piece is a quantitative attribute belonging to a particular decision alternative. Design decisions need to also consider qualitative attributes such as aesthetics. Human judgment is often required to make the appropriate trades without sacrificing the wholeness of the task.
- **Both use formal *models*** – Rules can be used as a model of some of object or concept. In multi-attribute decisions a utility function is a model of the

human decision-maker's value within a specific situation. In engineering design the model represents something more objective, such as a physical product that can be manufactured. In both cases, models provide useful information, but are not used as the sole basis for evaluation.

- **Both involve "*satisficing*"** - The exploratory nature of design and multi-attribute decisions means that it is difficult to determine the amount of effort that will be required in arriving at a satisfactory selection. More iterations generally improve decision-making, but this comes at a cost. However, neither situation is characterized by a well-defined stopping criteria. Exploration stops when the user is satisficed.

The above characteristics are common to both engineering design and multi-attribute decision-making, despite differences in their semi-Structured topology. Other decision systems that have the above characteristics tend to also have ill-defined aspects to their operational decision process.

3.7.7 Implications for Future CAD Tools

The paradigm for engineering design discussed in this section suggests that humans are most valuable for design conceptualization and evaluation—at each "end" of the decision process. Existing CAD systems use automation primarily in the "middle" of the decision process. In order to improve design through more automation, future CAD systems may push Structure outward towards each end. This is illustrated in Figure 3-45.

In design research, algorithms are currently being developed to *enhance* conceptualization and evaluation, which, although difficult, seems to be the appropriate use of additional Structure within the design process. For example, recent CAD research includes the generation of design concepts based upon ergonomic and aesthetic considerations, and the integration of data from multiple sources under a common design environment [112], [155]. Genetic algorithms can search huge (but well-defined) parameter spaces based on *a priori* fitness criteria [154] to produce, through evolution, a set of concepts that is of manageable size for applying human judgment. Semi-formal evaluation techniques such as Pugh charts, Quality Functional Deployment (QFD), and the Analytic Hierarchy Process, assist humans in *system* evaluation, but require subjective assessments at the component level. Axiomatic Design Theory goes so far as to provide a formal methodology for selecting "optimum" designs [146]. All of these techniques add Structure to the two "ends" of the design process, but appear to be most appropriately used as supportive tools rather than as human replacements.

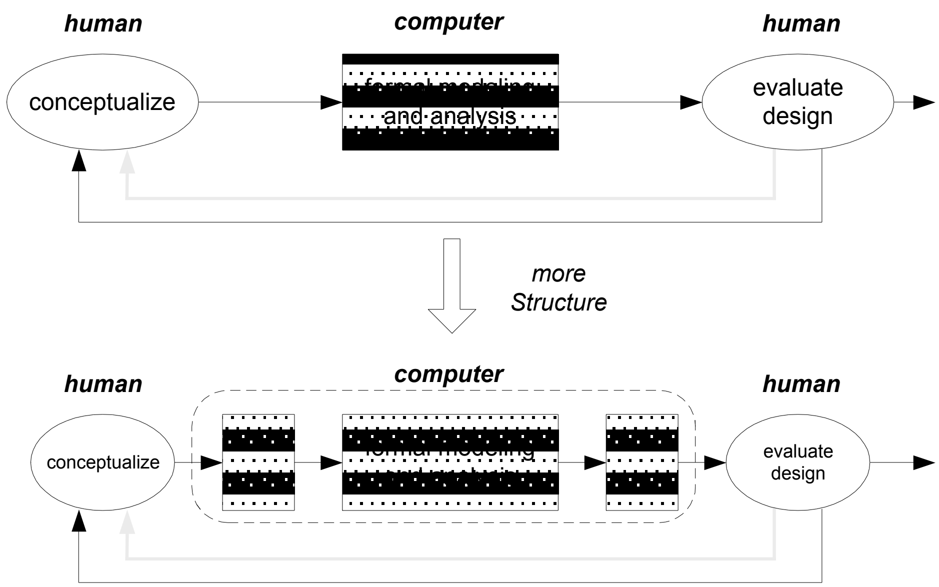


Figure 3-45 Future CAD may include more Structure in the "middle"

As research in engineering design continues, it is perhaps most important to recognize that ill-defined design goals and tacit knowledge form the fundamental problem in understanding conceptualization and evaluation. By ignoring this observation, efforts to improve engineering design through excessive use of Structure at each "end" of the decision process will often be misguided and, in fact, can be harmful.

3.8 CLOSING REMARKS ON EXAMPLE SYSTEMS

This chapter analyzed a variety of highly evolved decision systems—classified by their semi-Structured topology—that are representative of many other systems. It is assumed that these designs are "good" because they have evolved.

When a decision system is designed using the semi-Structured framework, there is a degree of freedom associated with how much of the decision process is Structured (versus Unstructured). The purpose of this chapter was to illustrate some of the manifestations of successful designs in order to provide insight, such as by identifying the parts of the operational decision process—the Unstructured parts—that are not fully determined at design time.

The purpose of this closing section is primarily to make some *general* observations from the collection of examples that were analyzed in this chapter. The discussion that follows includes a summary of why humans appear to be valuable in systems, a generalization of the information that is exchanged between Structured and Unstructured sub-processes, and a subsequent discussion on a fundamental dilemma in decision systems: resolving conflicts between multiple decision-makers.

3.8.1 The Value of Humans in Decision Systems

It appears from the examples that humans add value to Unstructured processes within a decision system. More specifically, the functions of humans reflect those discussed in section 2.7: "Reasons Why Structured Processes May Not Be Appropriate." Some of these are briefly reviewed:

- **Temperature Control –** Humans make subjective judgments, and transform a complex set of goals and information into a simple, well-defined goal: the temperature set point.
- **Aircraft Control** – Humans observe and integrate a complex set of environmental information as part of strategic, high-level control, and produce well-defined target states such as way points, altitude, and roll angle. Outside of nominal conditions humans are adaptive, and in safety-threatening situations will naturally use their survival instinct to find any means for

maintaining a safe flight. Humans are also responsible, and are thus held accountable for their decisions.

- **Diagnostics and Procedures** – Even when humans execute Structured processes in the form of standard operating procedures, Unstructured processes are used to initiate these rules. In the aircraft engine fire scenario, humans consider other information beyond engine temperature to verify/falsify the alert. In medical decisions, humans use complex pattern recognition skills to diagnose a malady, and consider the treatment procedure in a broad context before initiating it.
- **Multi-Attribute Decisions** – Humans are valuable in the decision loop for dynamic adjustments of goals and constraints based on the data set. They also are needed to make ill-defined trades and holistic evaluation of a set of distinct quantitative attributes.
- **Engineering Design** – Humans are valuable for understanding the design requirements, generating creative design concepts, and evaluating designs with respect to ambiguous goals. Evaluation requires integrating a complex set of information, and includes subjective judgment (e.g., aesthetics), moral judgment (e.g., economic-safety trades), and responsibility to society.

None of the examples in this chapter illustrated automated Unstructured processes such as neural networks. One reason is that these tend to be used in restricted domains, and are not common in highly evolved decision systems. Neural networks may offer value to some of the Unstructured elements in the above examples. In fact, fully automated semi-Structured processes, such as the marriage of neural networks and expert systems (sometimes referred to as "hybrid systems"), are well-known for their synergy [10], [57], [90]. Hence, Unstructured and semi-Structured processes can be valuable without humans. However, humans bring additional dimensions of value to decision-making, as they appear to be unmatched by their breadth of experience and their ability to incorporate "humanistic" requirements such as moral judgment.

3.8.2 Generalizing the Interaction Between Sub-processes

Some common characteristics of sub-process interaction can be identified from the variety of examples in this chapter. Specifically, while there are no limits to the detailed purpose of a sub-process, it is observed that decision systems often use sub-processes—Structured or

Unstructured—to generate well-defined information for other sub-processes in one of the following generalized ways:

1. *Provide Representation of System Goals* – Information that represents some component of the system goals is communicated to lower levels in a hierarchical decomposition, providing desired states for the controlled system. The transformation results in information that is simpler and less ambiguous than the actual system goals.

2. *Provide "Situation Assessment" Information* – Information that makes a decision process more informed, which includes observations of the environment, constraints on the solution process, models of decision consequences, etc. This includes any information that can affect the system decision outputs.

3. *Provide Representation of the System Decision* – Information that represents outputs of the process. That is, how to alter the controlled system. This information may be the result of a search process.

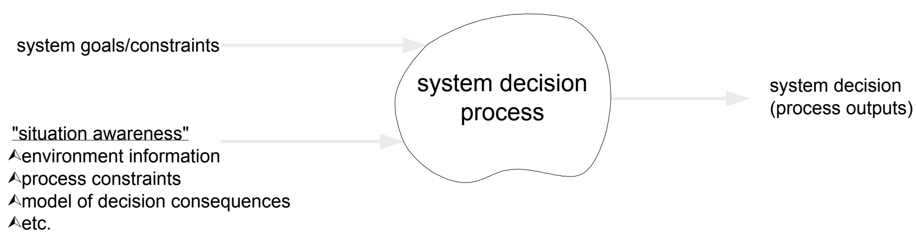


Figure 3-46 Generalized system inputs and outputs

The above information represents three fundamental types of inputs/outputs in a decision system, as shown in Figure 3-46. In other words, any designed system has a set of goals, a set of information that it processes during operation, and decision outputs. What is important to recognize here is that sub-processes are used within a system to provide information inputs to other processes that reflect these three categories—producing a representation that is well-defined. Hence, while the specific purpose of sub-processes within a system may vary greatly, their generalized purpose is limited in this sense.

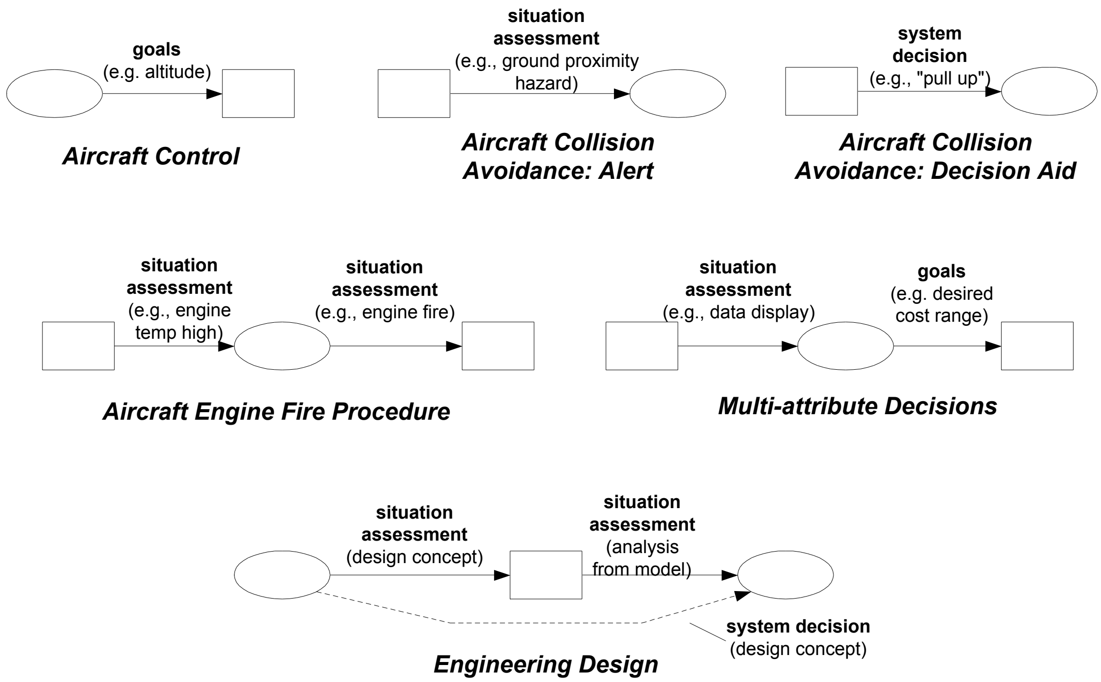


Figure 3-47 Generalized sub-process interaction (observed from example systems)

Figure 3-47 shows the example systems in this chapter, and identifies information between sub-processes as falling in one of the above three categories. The reasoning for these is explained as follows:

- **Aircraft Control** – Humans provide a well-defined *goal* to automation (e.g., altitude), which can be viewed as a formal representation of a more complex, ill-defined system goal.
- **Aircraft Collision Avoidance: Alert** – Automation provides *situation assessment* information for the Unstructured decision process. Computers process aircraft and environmental data and provide various representations of this data (e.g., graphical displays and alerts) to make the human's decision more informed.
- **Aircraft Collision Avoidance: Decision Aid** – Further along the spectrum of information automation, decision aids generate information that represent the *system decision* (and in the case of "auto-overrides" *act* execute this decision).

- **Aircraft Engine Fire Procedure** – Automation provides an alert for the human, who then considers other issues in an ill-defined manner before classifying the situation as an engine fire for the procedure. Both stages provide *situation assessment* for the following stage.
- **Multi-Attribute Decisions** – Three categories can be illustrated here. Data display provides *situation assessment to* the human. Humans provide attribute constraints and subjective weights that reflect the system *goals*. Also, not shown, the utility function represents a *system decision*.
- **Engineering Design** – Of the three sub-processes, "conceptualization" provides a representation of the *system decision* for the "evaluation" sub-process, but provides *situation assessment* information for the "analysis" sub-process. Furthermore, the "analysis" sub-process provides *situation assessment* information for "evaluation." This example shows that the classification of information depends on the process that *receives* the information.

It is observed from the examples in this chapter that, within a decomposed decision process, both Structured and Unstructured processes can provide information in the three categories: *system goals, situation assessment,* and *system decision*. These appear to be fundamental elements of designed decision systems that involve humans, and are not a direct result of a semi-Structured topology. However, it may be insightful to understand how Structure and Unstructure can both be used in support of other existing processes.

3.8.3 Resolving Conflicts in "System Decisions"

In the previous section, it was noted that the information communicated between sub-processes can sometimes be categorized as a *system decision*. In these cases, the information can potentially be used directly for decision-making, without further processing. The most common examples are decision aids. In decision aids, it is generally up to the human to determine how computer suggestions are used, particularly when the human is not in agreement with the decision aid. Outside the context of decision aids, a more fundamental question arises:

How can conflicts in *system decision* inputs be resolved?

That is, when multiple sub-processes within a system produce *system decision* information that are not identical, what is the basis for determining the actual system decision?

The conflict resolution problem is particularly critical in semi-Structured processes when system decisions are made by both the Structured and Unstructured sub-process. Since a semi-Structured decision does not have an objective *a priori* basis (such as an objective function), it is not possible to formally define which of the two is "better." The classical approach is to defer the conflict resolution decision to operation, in which case humans can use their judgment. In these cases, the information is used as a decision aid.

Aircraft Control Authority Revisited

Another approach is to resolve conflicts through a prescribed process. This was already seen in aircraft as an "auto-override," (Section 3.4.3) in which case automation seizes control of the aircraft in time-critical events for safety reasons (e.g., avoid wing stall). In the diagram of this process, the change in control authority was illustrated as a two-position switch, although the process that determined the switch position was not made explicit (Figure 3-48(a)). Figure 3-48(b) also shows the same decision process that illustrates the control authority decision process explicitly. The latter representation illustrates the fact that the decision of "who's in charge" is prescribed prior to operation.

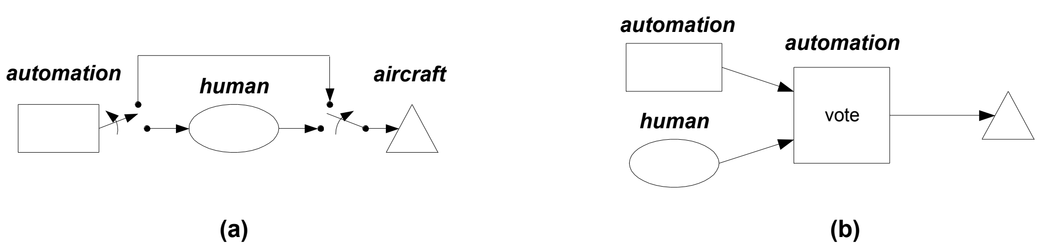


Figure 3-48 Implicit and explicit representations of the control authority decision process

The example in Figure 3-48 is a specific case of Structured conflict resolution, which can be labeled more simply as "voting." In the control authority example, voting was accomplished by ignoring the human based on a threshold condition. Voting can also be accomplished using other methods.

Other Structured Voting Examples

Figure 3-49 illustrates two other applications of Structured voting. In "mechanical voting," (Figure 3-49(a)) control authority is determined by the "strongest" actuator: human or machine. This method was used in early aircraft to allow humans to maintain control in the event of an actuator failure by designing the linkages between actuator and control surface to structurally fail based on an opposing human force. It illustrates a simple case of a more general hierarchical voting process, in which the rules determine which voter takes precedence.

In government elections or group decision making (Figure 3-49(b)), voting rules are typically not hierarchical, but collective. People use ill-defined processes to produce well-defined individual decisions (e.g., candidate choice), which are then integrated using a well-defined set of rules (e.g., majority vote) to produce a collective decision. The decision process of each individual is one sub-process within a larger system decision process, just as a decision-aid is one component in a larger human-automation system.

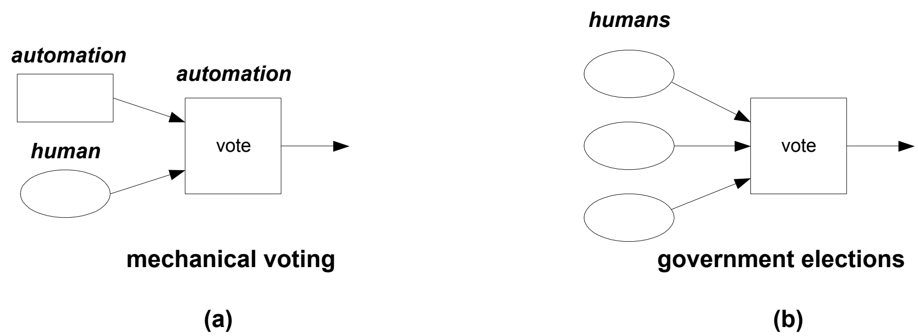


Figure 3-49 Other examples of Structured voting

All three cases in Figure 3-48 and Figure 3-49 resolve decision conflicts through rules, and produce an output that is identical to at least one of the inputs. Other Structured processes might instead *combine* inputs (if this is possible) using methods such as averaging. Whatever the specific method, resolving system decision conflicts from among multiple "voters" is frequently done through well-defined rules. However, a reasonable basis for Structured group decision-making is often elusive [7], [56].

Structured vs. Unstructured Conflict Resolution

The purpose of this section was not to answer the question of whether a Structured or Unstructured process is "better" at conflict resolution of *system decisions* from sub-processes. Rather, the point was to recognize that *system decisions* are an important category of information whose propagation to the system output ultimately has to be resolved through a dedicated decision process. It is up to the system designers to determine whether *system decision* conflict resolution is designed into the system as a prescribed process, or deferred to the operator. This design decision—Structured versus Unstructured—remains one of the most challenging in the design of human-automation decision systems.

CHAPTER FOUR

THE DESIGN OF A DECISION SYSTEM FOR A MOBILE ROBOT

4.1 INTRODUCTION

This chapter illustrates how the semi-Structured framework might be applied to the design of a decision system. While the framework does not provide a design methodology, it is useful to designers of decision systems because it provides a way to understand the implications of design choices, such as the allocation of functions between humans and automation. In particular, the framework forces designers to consider those parts of an operational decision process that are not completely determined at design.

A remotely operated mobile robot is chosen as a design example, in which decision system concepts are proposed and analyzed based on an assumed set of functional requirements. Emphasis is on identifying design trades rather than selecting a final concept.

4.2 USING THE SEMI-STRUCTURED FRAMEWORK IN THE DESIGN PROCESS

4.2.1 The Atypical Property of Decision System Design

Within the broad scope of engineering design, the design of decision systems is atypical. As discussed in Section 3.7, the engineering design process often terminates with a complete, explicit representation of the designed artifact. That is, before a design is manufactured and put to use, its form is fully determined. In contrast, the design of decision systems may be only *partially* determined prior to operation. The semi-Structured framework provides a way to consider the degree to which a decision system is determined prior to operation—that is, the degree to which a process is Unstructured.

4.2.2 Review of Decision System Design

As mentioned in Section 1.3.4, system-level design begins with a set of high-level sub-functions (the "whats"), which ultimately are translated into design concepts of decision processes (the "hows"). Since multiple processes can satisfy the same function, there are multiple ways to use humans and automation in a system. A system designer requires judgment to propose and evaluate design concepts as a set of interacting sub-processes. During the early stages in design, these concepts of decision processes are not yet in the detail of, say, computer code. However, the designer usually has sufficient knowledge to consider which of these sub-processes should perhaps be prescribed prior to operation, and which are better left determined until operation. The semi-Structured framework can help the designer better-understand the implications of design choices by helping to address questions such as:

- What is the purpose of sub-process interaction?
- What information is required by the different sub-processes?
- What are the interface requirements?
- What are the reasons why rules would not be appropriate?
- What value would the human bring to the system?
- How can Structure be used to support the Unstructured parts of a process?

In order to illustrate the application of the semi-Structured framework in a design problem, the following section considers the design of mobile robot decision system. Unlike in Chapter Three, this example system is not highly evolved. The purpose here is not to determine a "best" system design, but primarily to analyze the trades associated with different design concepts.

4.3 MOBILE ROBOT DESIGN PROBLEM

4.3.1 Mission Scenario

The primary mission is to perform remote situation assessment of an urban environment that may be occupied by enemy personnel. It is therefore primarily a scouting or reconnaissance mission—for providing information as to the presence and whereabouts of enemy personnel before sending in troops. Table 3 summarizes some of the key mission requirements. The mission is similar to other teleoperation scenarios such as planetary exploration, nuclear/chemical site inspection, or land mine detection, in that it is desirable to scout an area before ultimately sending in humans. In all of these cases, the operational environment is assumed to be complex, obstacle-rich, and not well known prior to operation.

Table 3 Key Mission Requirements

Primary mission	Remotely scout urban area for the presence of enemy personnel
Command/control range	10 kilometers
Area of scouting	0.1 square kilometers

The robot is assumed to be an unmanned *ground* vehicle (UGV). For this mission, the robot requires at least a camera for observing the local environment. Other sensors may also be needed, depending on the degree of autonomy chosen. At one end of the design spectrum, "manual" operation uses the human to perform all but the most trivial of functions on the robot, via a radio communications link—a situation that is similar to that of a toy remote-control car. However, a direct line-of-sight may not be possible, thereby requiring feedback to the remote human operator from artificial sensors. On the other end of the spectrum, "autonomous" operation requires that the robot have more-complex sensors and processing on board. Figure 4-1 illustrates a robot design concept for highly autonomous operation, in which some of its sensors are identified.

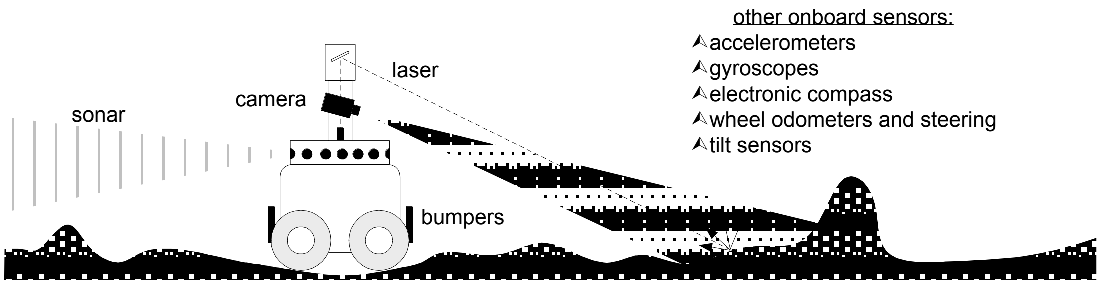


Figure 4-1 Example of highly autonomous robot hardware

Although the mission goals (system function) for the problem are well-defined, the means (operational decision process) for achieving these goals are difficult to prescribe prior to operation. A main reason for this is that the mission is *exploratory*, so that learning and adapting are inherent to decision-making. Furthermore, the operational environment can be difficult to interpret without human perception.

4.3.2 Functional Requirements

The requirements for this mission involve six primary sub-functions: situation assessment of a region within the larger area of interest, low-level motor control, terrain hazard detection, navigation, path planning, and system health monitoring. These are shown in Figure 4-2.

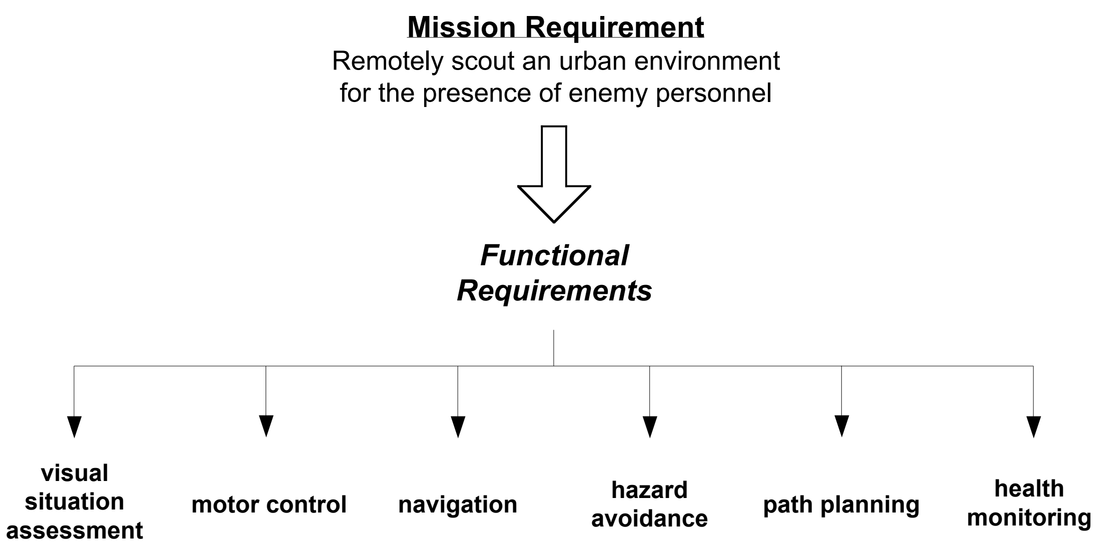


Figure 4-2 Robot functional requirements

Given the mission requirements and some initial assumptions of the robot configuration, the six sub-functions provide a framework for high-level system design. The sub-functions at this level allow a designer to consider the "whats" before diving into the "hows." The next section discusses design concepts—the "hows"—for each of the above sub-functions.

4.4 DESIGN CONCEPTS

This section analyzes different design concepts for satisfying the robot system function. "Designs" here refer to the high-level system concepts of *operational decision processes* and their allocation between humans and machines—not mechanical design, software design, etc. In order to facilitate the analysis, designs are discussed by considering each sub-function separately. Within each sub-function (numbered 1 through 6), different concepts are labeled alphabetically (e.g., *1-A*, *1-B*, etc.).

4.4.1 Visual Situation Assessment

The sub-function that is most closely related to mission requirement is the visual situation assessment of a local scene through a camera. The information available for this decision process is a series of images from the vantagepoint of the robot. For a given field of view, it is necessary to evaluate the scene for the presence of enemy personnel, and to determine their approximate location before sending in troops.

Concept 1-A: Autonomous Situation Assessment

The first design concept is to consider *autonomous* situation assessment. In this concept, an onboard camera can provide digital data for on board image processing.

Although the task does not fundamentally require humans, situation assessment is difficult in complex, exploratory environments. It remains difficult for machine vision to recognize other humans in these environments, particularly when they are not precisely known prior to operation. Unstructured approaches such as pattern matching with neural networks are not likely to work due to the lack of relevant training data.

Concept 1-B: Human Situation Assessment

Another design concept is to transmit raw video to the remote location of the human operator, allowing him or her to evaluate the scene perceptually (Figure 4-3). This design concept uses humans for image recognition, at what they naturally excel, but this comes at a cost in communications: on the order of 10 Mbits/second for video data rates. Such rates are considerably more than what is nominally required for command/control data.

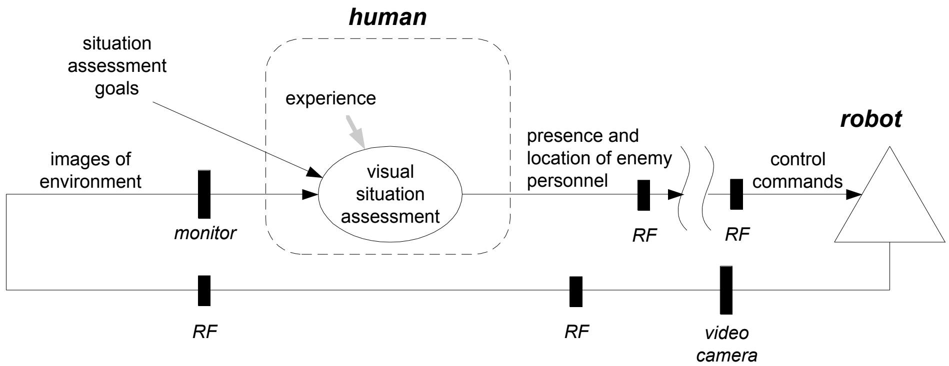


Figure 4-3 Human situation assessment (Concept 1-B)

The "RF" interface—shown in Figure 4-4 as a radio antenna and transmitter/receiver—between the control station and the robot is explicitly represented because it is a potentially critical element for operational decision-making. RF links are more susceptible to noise and interference than their hardwired equivalent, and more limited in the data rates they can support. It may be difficult to obtain a frequency band that is suitable for video data rates—particularly when line-of-sight communications is required.

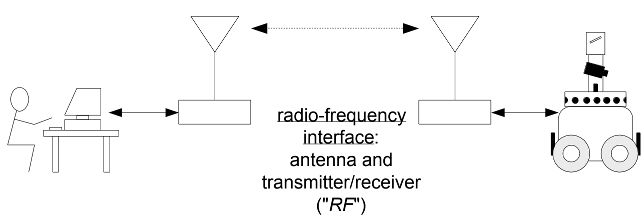


Figure 4-4 Radio "RF" communications link between operator and robot

An important reason for off-loading visual situation assessment to the human is because humans are good at recognizing familiar objects—including people—even within "noisy" environments. Because of their evolved perceptual processes and experience, such tasks are often trivial for people, despite the difficulty with machine vision. Furthermore, operators can easily learn from earlier parts of the mission (e.g., what typical backgrounds look like) and can improve as the mission progresses. Learning can be particularly important in exploratory environments, in which the terrain is unfamiliar.

Another important issue in situation assessment is responsibility. When a human is responsible for detecting enemy personnel, there is reason to believe that the task will be accomplished satisfactorily—which is critical due to the safety implications of the mission. Overlooking a well-camouflaged enemy may be difficult, but is not acceptable. By being held accountable for the task, humans may be motivated to perform situation assessment more thoroughly.

Concept 1-C: Visual and Infrared Image Fusion

Given the baseline design in which the human performs situation assessment, Structure can be added to enhance the image. For example, identical views from two cameras, each of which has a different spectrum of responsivity—visual and infrared—can be *fused* for enhanced observability in low-light conditions. This is illustrated in Figure 4-5.

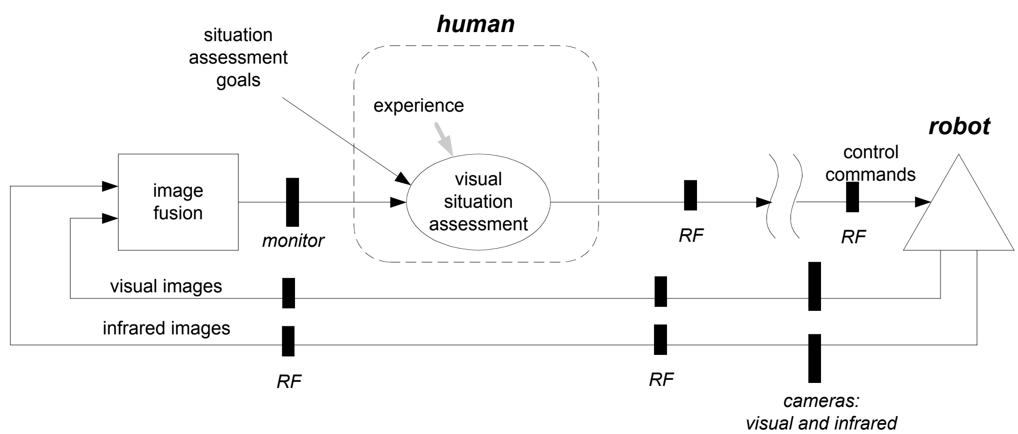


Figure 4-5 Structured image fusion to enhance human situation assessment (Concept 1-C)

While two sets of separate images may also be beneficial, image fusion integrates these into a single image based on well-defined rules. For example, the passive infrared detection of a human subject against a uniformly cold background does not provide references for location. This problem is eliminated when the subject is superimposed on the visual image. This concept illustrates that there is often some aspect of an Unstructured decision process that is sufficiently understood prior to operation such that Structure can be used to support the decision.

4.4.2 Motor Control

The function of motor control is to translate commands such as speed and heading into the appropriate actuator states (e.g., wheel torque, steeing position). These decisions must provide stable, efficient control under varying terrain conditions. Assuming that the robot is a ground vehicle that is limited to a speed of 1 meter/second, the actuators and their controllers can be limited to a bandwidth of only about 20 Hertz.

Since the environment is not explicitly incorporated into low-level control decisions during operation, the information required for motor control typically includes only target states and their associated feedback.

Concept 2-A: Autonomous Motor Control

Autonomous motor control (Figure 4-6) designs are highly evolved. There is little need to include the human in these operational decision processes, since the rules articulated in design have been shown to be robust over a large range of operational environments. Furthermore, if a human were in the loop, low-level control data would need to be transmitted in both directions in real time, adding to the communications requirements. Offloading low-level control—in which decisions are essentially continuous—to a remote operator can also introduce latency into the system, most likely from human information processing, but also potentially from some portion of the communications link.

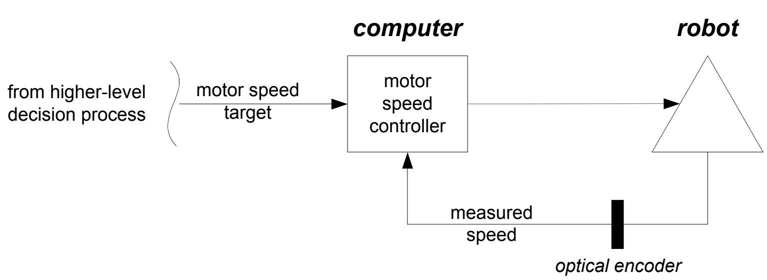


Figure 4-6 Autonomous low-level motor control (Concept 2-A)

Concept 2-B: Human Modification of Control Parameters

A modification of the standard design concept is to allow the human operator to adjust control gains during operation (Figure 4-7). Rather than develop a complex adaptive algorithm, humans can observe the operational environment from video images, and tune the robot's controllers based on expected and observed performance.

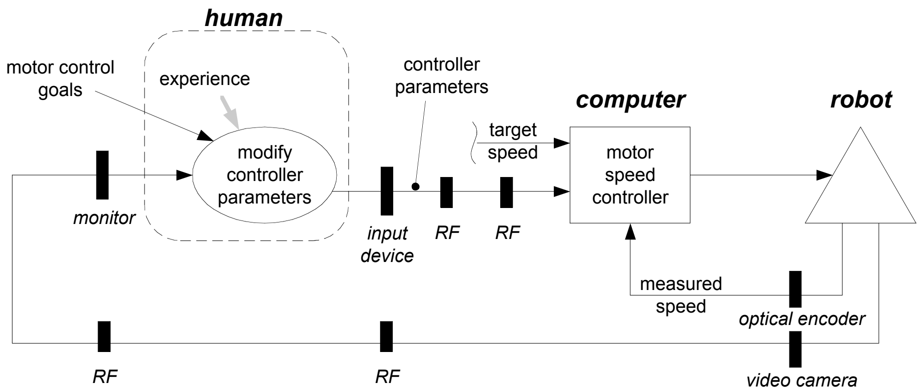


Figure 4-7 Modifying motor control parameters during operation (Concept 2-B)

By including a human in the operational decision process, the range of the motor controller can be extended. For example, controllers do not have to be compromised in order to be robust to all the conditions that may be expected. Instead, the operator can estimate parameter values that allow the controller to match the specific environment. The operator performs the complex role of understanding an image, and providing well-defined parameters for the Structured process.

4.4.3 Navigation

The function of navigation is to determine primarily the horizontal position of the robot within the area of interest—in this case the hostile urban environment. Requirements for this mission are a navigation *accuracy* of 3 meters, updated at one-second intervals. Position *resolution* is required to be 10 centimeters (a fraction of the robot size), updated at a rate of about 5 Hertz.

The information required for navigation is the robot position—or its derivatives—relative to some known reference. Since the reference of interest may be difficult to measure, *intermediate references* are often chosen, such as beacons, inertial frames, etc. This intermediate information must then be mapped to the appropriate frame.

It is assumed that the environment does not have a *local* navigation infrastructure, such as radio beacons or underground guide wires. Nor does the robot have access to a stored map or aerial image of the area. Therefore, local navigation references need to be obtained from other less-observable sources, or obtained from *global* references such as satellites.

Concept 3-A: Manual Navigation

One design concept is to determine the robot position within the urban environment by observing successive images from transmitted video. One benefit of this approach is that the robot does not require sensor hardware. However, as the robot's only source for position data, a human would be required to generate position updates at the required rate of 5 Hertz (assuming that humans are not controlling the robot manually, in which case navigation states can be implicit). Furthermore, humans may lose track of where they have been and where they are, particularly when visual cues are not sufficient.

Concept 3-B: Autonomous Navigation

A second design concept is to perform navigation autonomously. Autonomous navigation relies completely on onboard sensors after an initial reference near the beginning of the mission. The task of autonomously determining the position of the robot within its environment has become much more practical in recent years due to the decrease in cost, size, and power consumption of sensors whose data must be mathematically integrated in order to determine position. The onboard calculation of position can accommodate the required position update rates. However, the errors associated with these sensors can grow unbounded without a means for periodic calibration from references in the environment that provide a more direct means of obtaining position.

The operation of calculating position (relative to an arbitrary reference point) by mathematically integrating data is known as "dead reckoning." Sensors that provide dead reckoning data include wheel odometers and accelerometers. When these are "strapped down" to the robot (fixed relative to the robot), it is necessary to also provide orientation information—in this case *heading*—from sensors such as gyroscopes (which have to be integrated for *angular* position), magnetic compasses, star/sun detectors, etc. Hence, both translation and orientation sensors are required.

Dead reckoning sensors provide a means for calculating position using well-defined rules. However, the errors associated with measurements from intermediate references cause small errors to translate into potentially large position errors. For example, accelerometer bias errors result in robot position errors that grow quadratically with time. In practice, ground vehicles (opposed to air or sea vehicles) do not typically rely on the integration of accelerometers, since they have the advantage of another stable reference: the terrain. However, odometry is also error-prone when its variations are not known prior to the mission. Due to dead-reckoning errors, it is therefore necessary to periodically calibrate robot position during the course of a mission.

Periodic calibration during a mission is hard to achieve autonomously without a local navigation infrastructure. Global infrastructures such as the Global Position System (GPS) do not provide the necessary accuracy of 3 meters, and are not designed to work indoors—a situation that may arise during a mission. Furthermore, the recognition of landmarks through machine vision is difficult. In short, autonomous navigation can be performed through dead reckoning, but is limited in this mission scenario due to the need for periodic position calibration. This limitation can be overcome by including humans in the navigation decision process.

Concept 3-C: Manual Calibration

A third design concept for the navigation process exploits advantages of each of the above concepts. This uses the human for periodically calibrating the robot position and orientation using images of the environment, and retains automation for continuous position calculations through dead reckoning. The human operator essentially acts as an Unstructured observer of navigation references that are otherwise ambiguous. This is illustrated in Figure 4-8, in which dead reckoning is performed autonomously based on data from a heading gyro and odometer.

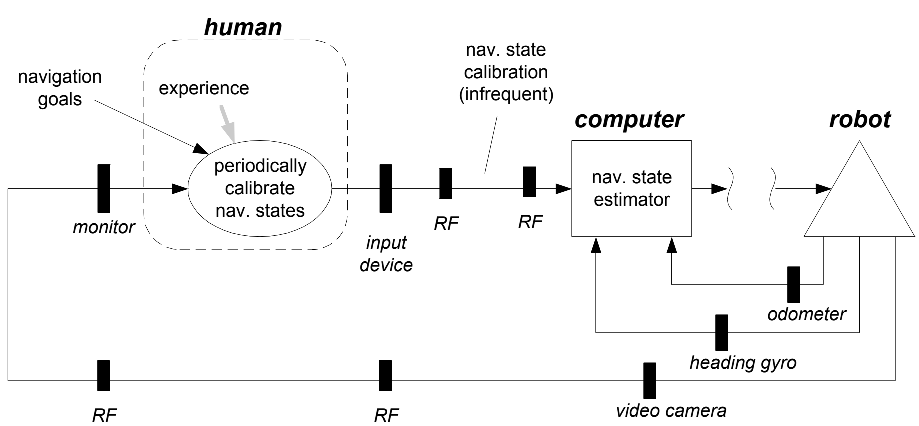


Figure 4-8 Manual calibration of navigation states (Concept 3-C)

One of the reasons why humans are able to use visual images for calibration is because of their familiarity with objects in the environment. Not only can they recognize objects from different vantage points, but they know the typical sizes of cars, sidewalks, buildings, trees, etc., and can gage their distance partly based on their size. These abilities allow humans to estimate position without explicit calculations. At the very least, far away objects may be used to calibrate the robot's heading.

Since humans are not required to calibrate frequently, this design strategy is more practical than a fully manual approach. However, a critical design issue is determining the representation of the calibration data, since a transformation is required between the ill-defined perceptual domain of the human and the precise mathematical coordinate frame of the robot.

4.4.4 Hazard Avoidance

The function of hazard avoidance is to detect and avoid mobility hazards in the environment. This involves local, temporary trajectory modifications based on current or projected hazardous states.

A mobility hazard can take many forms. The most common hazard is an obstacle that cannot be surmounted. Other hazards include drop-offs, craters, soft soil or mud, water, etc. When a robot encounters these, there is an increased chance of mission failure from decreased mobility. Depending on the environment, hazards may be ambiguously defined. For example, an area cannot typically be clearly decomposed into only "hazardous" and "safe" areas.

While it is often desirable to detect hazards *before* they are directly encountered, it is sometimes possible to also deal with hazards upon contact. These two situations are reflected by the basic information required for hazard-avoidance decisions: terrain profiling (e.g., range measurements) and the current mobility hazard state of the robot (e.g., collision, tilt).

In addition to observing hazards, the hazard avoidance decision system must also make trajectory modifications to circumvent them. This level of maneuvering is considered lower level than "path planning," since the avoidance maneuver is a temporary deviation from the primary path.

Concept 4-A: Manual Hazard Avoidance

The first design concept utilizes the human for detecting and avoiding hazards based on sensor feedback from video, tactile, and tilt sensors. Like manual navigation, evaluation of data must be performed at a remote location at the cost of communications.

The advantage of this concept is that humans can accommodate the ambiguity of hazards, and can exploit their perceptual processes and experience for detecting them. Given sufficient information, humans can use their broad knowledge to make a more accurate assessment of a hazard, particularly when hazards are ambiguously defined.

The *remote* evaluation of hazards can also be difficult for humans. Sensory feedback that would otherwise be perceived directly must now be perceived through artificial sensors. Video is perhaps the most natural representation of information, but is limited in its field-of-view and is a medium from which it is difficult to extract depth (without stereo imagery). Other sensor data may be difficult to monitor remotely because of the non-intuitive form relative to direct perception (e.g., tactile). Lastly, hazards such as collisions cannot afford the delays associated with offboard sensor processing and control responses.

Concept 4-B: Autonomous Hazard Avoidance

A second design concept is to perform all hazard avoidance decisions onboard the robot. This eliminates communications and latency issues, but generally requires more-elaborate sensors and processing.

At the simpler levels of hazard avoidance, threshold-based hazard detection from tactile and tilt sensors are used to generate simple control maneuvers. For example, when a collision is detected through single-threshold tactile sensors, a typical maneuver is to immediately stop—which is designed to prevent the condition from worsening—and then to travel in reverse or retrace its previous path a short distance, followed by an alternate forward route to the left or right. These actions may not be "optimal" in all conditions—like an alarm, their simple logic can produce false alarms and missed detections. However, these rules allow the robot to potentially circumvent hazards without human assessment.

Autonomous hazard detection within a complex environment requires sensors to explicitly measure portions of the terrain before its contact with the robot (refer to Figure 4-1). This is required primarily because terrain mapping is difficult to perform with machine vision. Active range finding is used to calculate the distance to the terrain based on emitted energy in the form of light (e.g., laser) or sound (e.g., sonar) in a narrow field of view. Because of the emitter's narrow field-of-view, scanning is required to map the robot's surroundings. Laser range finding has the advantage of providing a detailed map, but the mechanical scanner adds considerable complexity. A low-cost alternative is to use a static array of ultrasonic range finders, which provide broad coverage at a cost of resolution in terrain mapping.

One of the drawbacks to autonomous hazard avoidance is the sensor suite that may be required to provide adequate observability. More fundamentally, the effective fusion of data from multiple sensors remains difficult. In fact, humans or neural networks may be a better choice in these situations since the relevant information is available, but the rules for detection and avoidance do not have to be explicitly articulated.

Concept 4-C: Human Modification of Hazard Data

A third design concept is to use humans as an Unstructured observer for automation, identifying ambiguous areas as safe or hazardous, or overriding the robot's observations. This design *augments* the autonomy, extending its range of operation. Figure 4-9 illustrates this concept.

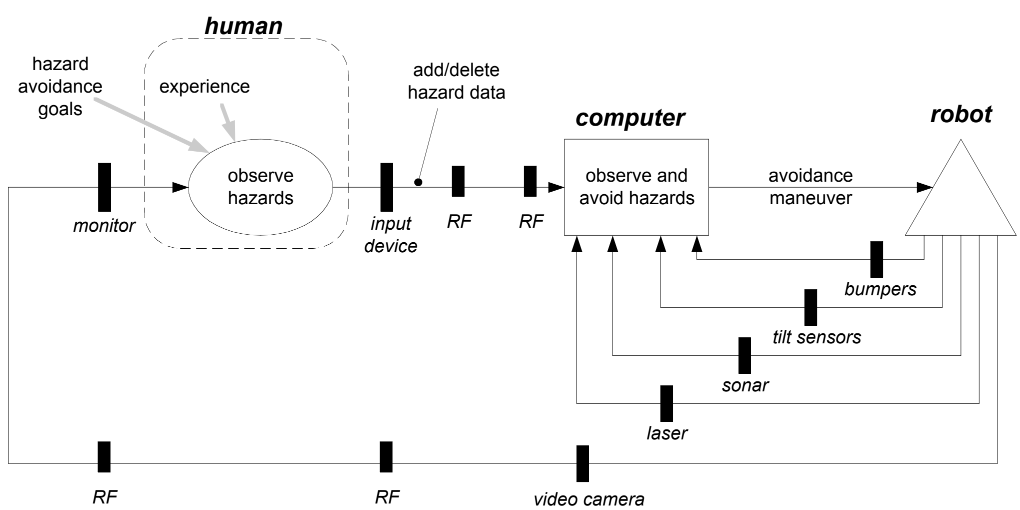


Figure 4-9 Modification of hazard data (Concept 4-C)

As an example, sonar suffers from "specular" (opposed to "diffuse") reflections when the ultrasonic beam hits walls at grazing angles. Due to the relatively large wavelength of ultrasonic energy—on the order of a millimeter, versus a micrometer for light—many surfaces appear mirror-like. Humans can identify these surfaces from video, and provide inputs to automation about their location. Other hazards within the environment such as water, mud, and energy-absorbing material (e.g., flat black surfaces absorb light) may be easily perceived from video images, enhancing the system's situation awareness. Even collisions that are missed by bumpers (it is difficult to cover a significant portion of the robot with tactile sensors) may be observable from video. Similarly, humans can identify non-hazards such as tall grass, empty boxes, or other objects whose presence is detectable through range finding, but does not pose a hazard to the robot.

This design concept requires a means for the teleoperator to add or delete hazard data that is detected by automation. Although the objects may be easily apparent to the human, the information needs to be transformed to the appropriate representation for automation. For example, if an object is selected by outlining a portion of the image on the monitor—which is intuitive—software is needed to translate this area to the robot's three-dimensional map. A simpler but less-intuitive approach is to force the human to specify the objects in the robot's coordinate frame. In either case, the interface at the control station needs to allow the human to communicate this information.

Concept 4-D: Fusion of Video and Range Data

Another design concept that builds upon the previous design is to use automation as an interface: to provide a more-natural representation for both observing hazard information, and for communicating hazard information to the robot's computer. This is shown in Figure 4-10.

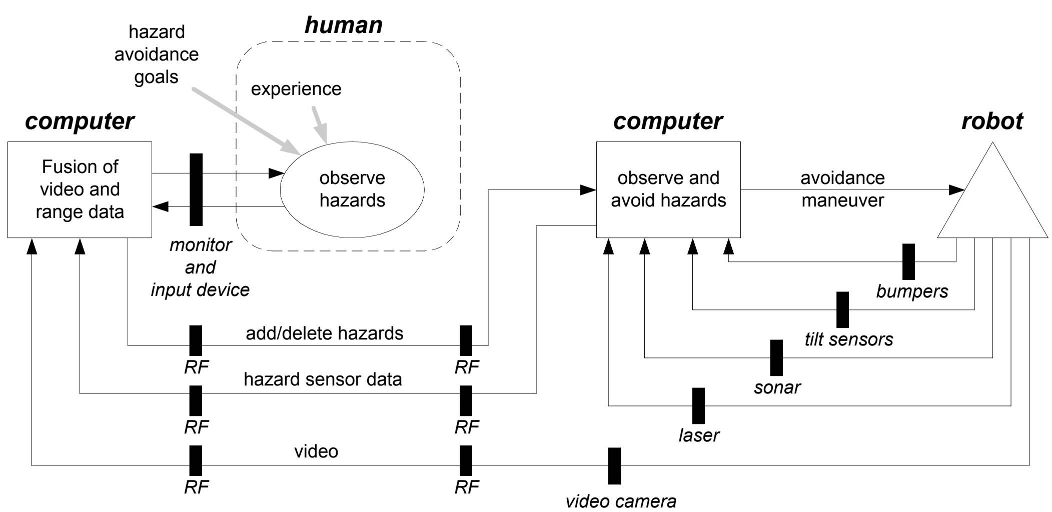


Figure 4-10 Structured data fusion for observing and modifying hazard data (Concept 4-D)

The primary change from Figure 4-9 is the addition of the Structured process at the left. By fusing video with range data using well-defined rules, the operator can better understand the local environment. Furthermore, when objects in the enhanced image need to be modified (added or deleted), the display can provide an intuitive graphical representation for the human to identify objects. This information can then be transformed to the coordinate frame used by autonomous hazard avoidance using well-defined rules.

4.4.5 Path Planning

The function of path planning is to generate a long-term route for the robot. However, due to the exploratory nature of the mission, the path cannot be completely prescribed prior to operation, and will likely require modifications during operation. Aside from temporary deviations due to hazard avoidance maneuvers, a path represents the highest-level mobility goal for the robot. The information for path planning includes the mission goal, a representation of the area, a terrain profile, and information from situation assessment—which evolves as the mission progresses.

Since path planning is high-level, it can typically be commanded hierarchically in an under-constrained representation. In other words, the path can be specified *generally* at one level, and determined in *detail* at lower levels. One way to accomplish this is through waypoints, which constrain a path to discrete locations, but do not determine the path outside of these constraints. Waypoints represent intermediate goals which collectively "pull" the robot across the terrain in the desired manner.

Concept 5-A: Autonomous Path Planning

One design concept is for the robot to plan its path autonomously. Autonomous path planning has been the focus of mobile robot research for years, and many algorithms have been developed with varying degrees of success. However, it is often the difficulty of *this* task that has limited the autonomy of robots in uncertain, ambiguous operational environments, which is the case here.

Autonomous path planning for mobile robots has been successful primarily in laboratories and other well-defined environments, in which obstacles are clearly represented and known with certainty prior to operation. In most of these cases, regions of the environment can be represented as one of two states: obstructed or clear. Figure 4-11 illustrates this situation for two waypoints. In this representation, paths can be generated based on techniques such as constrained optimization (e.g., minimize fuel), in which obstacles are incorporated into the decision process as constraints.

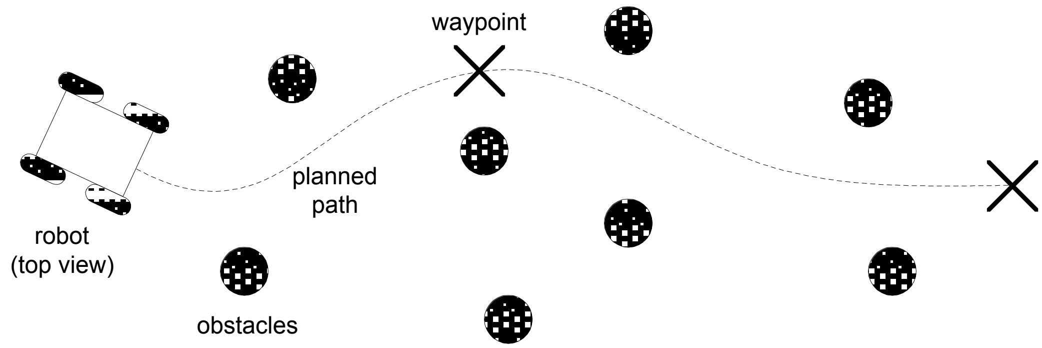


Figure 4-11 Path planning in a well-defined environment

In urban environments, the representation in Figure 4-11 may not be appropriate for path planning. The terrain has properties—both geometric and mechanical—that lead to ambiguous hazard states. That is, hazards cannot be adequately represented with one bit of information, and may be difficult to represent at all. However, even with a more-complex representation of the environment, the exploratory nature of the mission prevents the environment from being known with a high degree of certainty. In order to plan strategically, waypoints need to be generated based on incomplete, ambiguous, and uncertain information about the environment and the people within.

Concept 5-B: Human Supervisory Path Planning

Another design concept is to use the human in the role of a supervisor. In this strategy, the operator issues waypoints, but lets automation determine the remainder of the path. Also, the human monitors path execution as it unfolds, and can intervene to seize control at all times. An illustration of this concept is shown in Figure 4-12.

Again, the human relies on video for the ill-defined observation of the environment. Humans are often good at long range planning based on limited short-term information. That is, they can infer larger portions of the environment from what is directly observed (e.g., understand that a room lies behind a door). Since high-level path planning requires a bigger picture view of the environment, local observations without such inference are insufficient. Humans are valuable because these judgments are made using ill-defined processes.

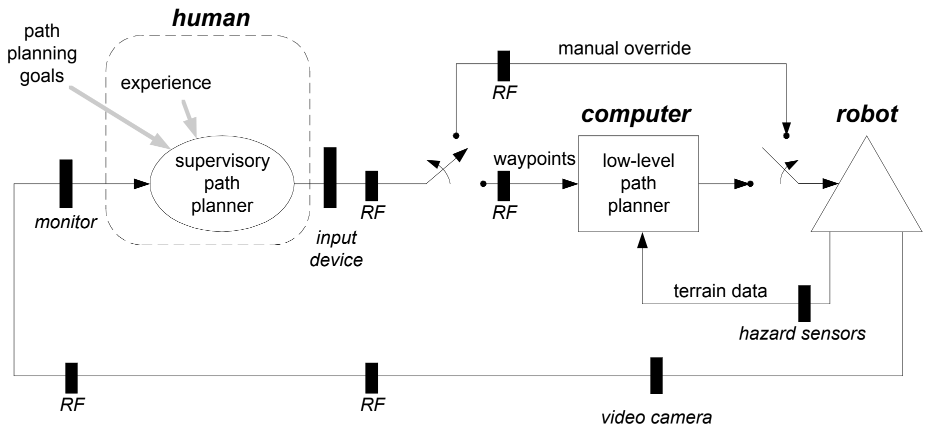


Figure 4-12 Human supervisory path planning with manual override (Concept 5-B)

In this design, humans do not specify the entire path, but only issue waypoints (in nominal conditions). The generation of these waypoints may consider a host of complex, ill-defined goals and information, but the waypoints themselves are simple and well-defined. Automation then uses these as constraints from which to calculate optimal paths based on criteria such as fuel or distance. If necessary, humans can also have direct control of this path through manual overriding the low-level path planner. This gives the operator the means by which to have complete control authority and therefore responsibility of the robot.

4.4.6 Health Monitoring

The purpose of health monitoring is to observe various robot states for maintaining system functionality. When anomalies such as failures are detected, the system may be reconfigurable in a way that allows the mission to continue. For example, if a drive motor fails and is detected, a reconfiguration may be to disconnect that motor from the drive train, allowing it to spin freely. Without at least *observing* a threatening health condition, the success of the mission can be jeopardized. This is why health *monitoring* is a critical function.

The information required for health monitoring can include any robot state, from low-level internal states (e.g., motor current) to high-level performance states (vehicle speed). Since the definition and number of "states" is arbitrary to a certain extent, "good coverage" is more practical than theoretical. That is, since a system is never completely observed (even the addition of a health monitoring system creates more states to observe) design choices must be made to determine which states are to be explicitly measured. Fortunately, since some failures propagate or emerge in predictable ways, it is not always necessary to measure states directly, but to infer them from observed symptoms. This reduces the complexity of the health monitoring hardware, but it relies on a stricter set of assumptions.

Concept 6-A: Automated Decision Aids

One design concept is to use automation primarily as decision aids, and the human for making the ultimate health diagnosis. When failures are not well-modeled, humans can use reasoning to estimate internal states from observed symptoms—some of which may not be explicitly measured. Unmanned ground vehicles have the advantage of being able to temporarily shut down systems—including the drive motors—while offline diagnosis is performed. This may not be possible in other missions, due to the *immediate* need to reconfigure (for example, communications satellites cannot afford to have downtime, and unstable aircraft cannot afford to lose flight control temporarily).

Given that the system can afford temporary downtime, humans are used for remote diagnosis primarily because of the limited amount of onboard sensing. The symptoms that emerge from anomalies can be used to infer possible internal states when these states are not explicitly measured, as in medical diagnosis. However, alerts and decision aids can assist humans, using formal inference rules (e.g., expert systems), neural network pattern recognition, and simpler logic from direct sensing. These are illustrated in Figure 4-13. The diagnosis algorithms can be more complex if the executed at the control station. However, this requires that raw sensor data be transmitted. Offboard diagnosis may also introduce latency.

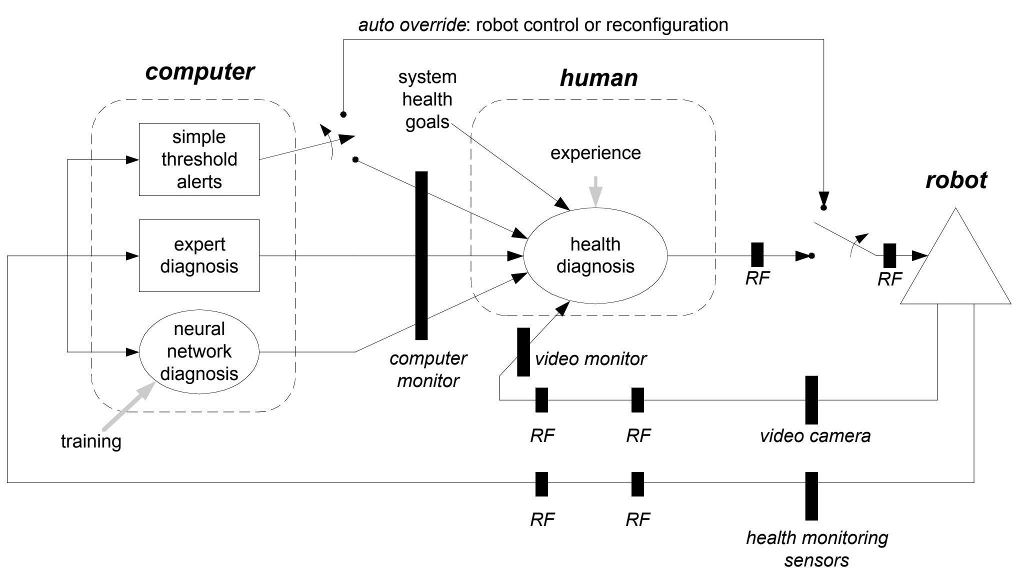


Figure 4-13 Automated health monitoring and reconfiguration (Concept 6-A)

Despite the complexity of a robotic system, some portions are understood sufficiently to warrant immediate autonomous action. For example, when a temperature sensor indicates that a drive motor is overheating, the robot can be automatically shut down to avoid permanent damage. Since damage can occur rapidly, and since there is little cost associated with temporarily shutting down, auto-override is justified. Due to the chance of a communications failure, decisions such as these should be implemented onboard the robot.

4.5 ROBOT DESIGN SUMMARY

This section examined design strategies for a robotic mission in a hostile urban environment. These designs represent the operational decision process for the human-automation system. Although the mission has relatively well-defined goals, there are many aspects of the operational decision that are not understood during design, such that portions of the decision process are deferred to operation. The semi-Structured framework was used to help guide design—to understand the limitations of *a priori* rules, thereby using humans and automation appropriately within the operational decision process.

In nearly all six sub-functions, the human is critical in the decision process. An understanding of the ill-defined aspects of the goals, as well as experience, are important "inputs" for operational decisions. Video feedback from the robot provides images of the robot's environment to support these decisions. Data from other robot sensors is also valuable. Since the operator is at a remote location, all operational data observed by the human is obtained through communication from the robot, and therefore available to automation as well. Despite that automation has access to the same robot data, many of the tasks are inappropriate for automating.

Table 4 summarizes the tasks of the human operator and automation within the robot's operational decision system. In some of the sub-functions—situation assessment, path planning, and health monitoring—the human plays a dominant or supervisory role. For these, automation enhances human decision making by providing information (*Structured into Unstructured*), or acting in a subservient manner based on well-defined goals from the human (*Unstructured into Structured*). In other sub-functions—motor control, navigation, and hazard avoidance—automation plays a more dominant role. In these cases, the human acts primarily as an observer for automation by providing valuable information that cannot be obtained directly from onboard sensors. Since the cost of information from sensors onboard the robot can be high, using the human in the decision loop can potentially reduce this cost.

Table 4 Summary of operational tasks for human operator and system computers (robot and control station)

Sub-function	Human Task(s)	Automation Task(s)
Visual Situation Assessment	Evaluate video for presence of enemy personnel	Image fusion of visual and infrared data
Motor Control	Modify control parameters	Low-level feedback control
Navigation	Periodically calibrate navigation states for the position estimator	Continuous "dead reckoning" position estimation
Hazard Avoidance	Add/delete hazards for automation	Sensor-based hazard avoidance; video/range data fusion
Path Planning	Issue waypoints; override robot path control	Optimal path control based on waypoint constraints
Health Monitoring	Robot system diagnosis	Diagnostic decision aids; auto-override of control or reconfiguration.

In this design scenario, the distribution of work between humans and automation has considerable implications on the communications system. The design decision to offload work to the human operator translates to the need for a radio link to provide the necessary information for the task decision. Hence, the cost of the interface—power, packaging, noise, bandwidth, etc.—is of particular concern here. Compared to missions such as planetary exploration, in which communications are more of a design driver, this mission can make ample use of humans in the operational decision process. Otherwise, emphasis would likely shift towards more automation onboard the robot. Adding more Structure to the decision system must be done carefully, however, since removing the human from the operational decision process requires more assumptions about the mission goals and the operational environment.

CHAPTER FIVE

SUMMARY AND CONCLUSIONS

This thesis introduced a novel framework for analyzing human-automation decision systems, based on the concept of the "semi-Structured" process. The purpose of this framework is to provide insight into decision systems, and ultimately improve their designs—in particular the allocation of functions between humans and automation.

5.1 DECISION SYSTEM DESIGN ISSUES

In this thesis, an important distinction is made between two time periods: the period of *design*, and the period of *operation*. This distinction is important because physical artifacts are often *completely specified* during design. Hence, any inaccurate or missing information during design regarding the operation of the artifact (e.g., system goals, assumptions about the environment) can compromise functionality.

The design of decision systems differs from most general engineering design because parts of the designed system do not have to be completely specified during design. These parts (sub-processes) can be deferred to operation. This property of decision systems can be advantageous to the designer because design assumptions made about operation can be relaxed, but it adds uncertainty to *how* operational decisions will be made.

An important design choice is determining the extent to which a decision system is specified prior to operation. A primary goal of this thesis was to provide a way to understand the issues associated with this design choice.

5.2 REVIEW OF APPROACH

The approach taken in this thesis was to consider operational decision systems as information processes. Given a system function, which describes the "whats," a process describes "how" this function might be achieved. The "semi-Structured" process was introduced as a way to consider both the well-defined and ill-defined parts of a decision system prior to operation. This is defined as follows:

> **Semi-Structured process** – A system of Structured and Unstructured sub-processes
>
> - **Structured process** – A process that can be reduced to well-defined rules
> - **Unstructured process** – A process that *cannot* be reduced to well-defined rules

These simple definitions are intended as conceptual tools to help understand decision systems. They reflect the extent to which a decision process is *explicitly* understood. Structured processes (such as traditional algorithms and procedures) capture the declarative knowledge of precisely how a decision is to be made, while Unstructured processes reflect where this type of knowledge is missing. While some non-traditional algorithms such as neural networks can be considered Unstructured, it is believed that humans primarily add value to Unstructured processes, while Structured processes can often be reliably automated.

In the context of decision system design, Structured sub-processes are the parts that are completely prescribed prior to operation, while Unstructured sub-processes are the parts that are left undetermined until operation. In other words, the Structured parts of a process are where the designer has removed decision-making authority from the operator, while the Unstructured parts are where the operator is given authority in determining how the decision is to be made. Structured processes provide a way for designers to exploit what is explicitly understood prior to operation, but they are also limited by what is *not* understood. While Unstructured processes may move the system away from optimality during nominal conditions, and add a degree of uncertainty to *how* a decision will be made, they may also add flexibility and robustness. It is ultimately up to the judgment of the designers to determine the extent to which a process is Structured—the extent to which a system is determined prior to operation.

5.3 CONCLUSIONS

One of the main contributions of this thesis was to elicit a way for people to think about the ill-defined "Unstructured" parts of a decision process in addition to the well-defined "Structured" parts. The semi-Structured framework includes consists of definitions, diagrammatic notation, and organizing principles. The goal was not to provide a methodology for design, but was primarily to provide insight into decision systems, and to improve upon *ad hoc* design approaches.

The semi-Structured framework is particularly valuable for explicitly identifying those parts of the operational decision process that are not understood prior to operation. That is, the framework provides a way to explicitly consider, to a certain extent, those parts of the process that are to be *fully* considered at a later time—typically with human operators. Without this analytical tool, the ill-defined components of a decision process may be overlooked or not incorporated effectively into a system design. By considering parts of the decision process as Unstructured, the system may then be designed more appropriately—such as by adding Structure to support rather than replace these ill-defined components.

REFERENCES

1. Ackoff, R.L., 1979, "The Future of Operational Research is Past", *Journal of the Operational Research Society,* Vol. 30, No. 2, pp. 93-104.

2. Ackoff, R.L., 1977, "Optimization + Objectivity = Opt Out", *European Journal of Operational Research,* Vol. 1, pp. 1-7.

3. Adams, M. and Hansman, R.J., 1991, "Last Hurdle for Autonomous Air Vehicles", *Aerospace America,* pp. 28-31, Oct. 1991.

4. Agre, P.E., 1988, "The Dynamic Structure of Everyday Life", Ph.D. Thesis, MIT.

5. Andre, A., and Degani, A., 1996, "Do You Know What Mode You're In? An Analysis of Mode Error in Everyday Things," *Human-automation Interaction*, M. Mouloua and J.M. Koonce (Eds.), Lawrence Erlbaum, pp.19-28.

6. Antunes, C.H.; Alves, M.J.; Silvas, A.L.; and Climaco, J.N., 1992, "An Integrated MOLP Method Base Package - A Guided Tour of TOMMIX," *Computers and Operations Research,* Vol. 19, No. 7, pp. 609-25.

7. Arrow, K.J., 1963, *Social Choice and Individual Values,* 2nd ed., John Wiley and Sons, New York.

8. Ashby, W.R., 1956, *An Introduction to Cybernetics,* Methuen.

9. Bailey, R.W., 1982, *Human Performance Engineering: A Guide to System Designers,* Prentice Hall, New Jersey.

10. Barker, D., 1990, “New Partners in the AI Dance: Neural Networks and Expert Systems,” *AI Week*, Vol. 7, No. 9, pp. 1-6.

11. Bateman, D., 1994, "Ground Proximity Warning System (GPWS) - Success and Further Progress," *The International Civil and Military Avionics Conference,* London. April 7.

12. Beach, L.R., and Lipshitz, R., 1993, "Why Classical Decision Theory is an Inappropriate Standard for Evaluating and Aiding Most Human Decision Making," *Decision Making in Action: Models and Methods*. G.A. Klein, J. Orasanu, R. Calderwood, and E. Zsambok (Eds.), Ch. 2, pp. 21-35.

13. Belton, V., and Elder, M.D., 1994, "Decision Support Systems: Learning from Visual Interactive Modeling, *Decision Support Systems,* Vol. 12, pp. 355-64.

14. Billings, C.E., 1996, "Some Questions About Advanced Automation," *Human-Automation Interaction*, Mouloua, M. and Koonce, J.M. (Eds.), Lawrence Erlbaum, pp. 314-20.

15. Billings, C.E., 1996, *Human-Centered Aviation Automation: Principles and Guidelines,* NASA Technical Memorandum 110381.

16. Blanchard, B.S. and Fabrycky, W.J., 1981, *Systems Engineering and Analysis,* Prentice-Hall, New Jersey, pp. 2-15.

17. Boy, G.A., 1998, “Cognitive Function Analysis for Human-Centered Automation of Safety-Critical Systems,” *CHI 98*, 18-23 April 1998, pp. 265-72.

18. Bradley, S.R., and Agogino, A.M., 1993, "Computer-Assisted Catalog Selection with Multiple Objectives," *ASME Design Theory and Methodology 1993*, DE-Vol. 53, pp. 139-47.

19. Bringsjord, S., 1998, "Chess is Too Easy," *Technology Review,* March/April 1998, pp. 23-28.

20. Brooks, R., 1986, "A Robust Layered Control System for A Mobile Robot," *IEEE Journal of Robotics and Automation,* March, 1986.

21. Bucciarelli, L.L., 1994, *Designing Engineers,* The MIT Press.

22. Buchanan, J.T., Henig, E.J., and Henig, M.I., 1998, "Objectivity and Subjectivity in the Decision Making Process," *Annals of Operations Research,* Vol. 80, pp. 333-45.

23. Cleveland, W.S., 1988, *Dynamic Graphics for Statistics,* Wadsworth and Brooks, California.

24. Conrad, M. and Rahimi, M.A., 1984, "Computers and the Future of Human Creativity," AFIPS Conference Proceedings of the 1984 National Computer Conference, Las Vegas, NV, July 1984, AFIPS Press. pp. 461-7.

25. Craik, K.J.W., 1947, "Theory of the Human Operator in Control Systems: 1. The Operator as an Engineering System," *British Journal of Psychology,* Vol. 38, pp. 56-61.

26. Dawes, R.M., 1982, "The Robust Beauty of Improper Linear Models in Decision Making," *Judgment Under Uncertainty: Heuristics and Biases*, Kahneman, D., Slovic, P., and Tversky, A. (Eds.), Ch. 28, pp. 391-407.

27. De Neufville, R., 1990, *Applied Systems Analysis,* McGraw-Hill, New York.

28. DeCelles, J.L., 1991, "The Delayed GPWS Response Syndrome," Aviation Research and Education Foundation. Herndon, VA. July.

29. Dehaene, S., 1997, *The Number Sense: How the Mind Creates Mathematics,* Oxford University Press, New York.

30. Dennet, D.C., 1971, "Intentional Systems," *Journal of Philosophy,* Vol. 68, No. 4, pp. 87-106.

31. Dershowitz, A., 1998, "The Effect of Options on Pilot Decision Making in the Presence of Risk," MIT Ph.D. Thesis.

32. Descartes, R., 1637, *Discourse on the Method for Rightly Conducting One's Reason and for Seeking Truth in the Science.*

33. Devlin, K., 1999, *Infosense: Turning Information into Knowledge*, W.H. Freeman and Co., New York.

34. Donoho, A.W., Donoho, D.L., and Gasko, M., 1988, "MacSpin: Dynamic Graphics on a Desktop Computer," *IEEE Computer Graphics and Applications,* July 1988, pp. 51-58.

35. Drake, A.W. and Keeney, R.L., 1992, *Decision Analysis,* 6th printing, Video Course Manual, MIT Center for Advanced Engineering Study, No. 37-2100.

36. Dreyfus, H.L., 1997, "Intuitive, Deliberative, and Calculative Models of Expert Performance," *Naturalistic Decision Making*. Klein, G. and Zsambok, C.E. (Eds.), Ch. 2, pp. 17-28.

37. Dreyfus, H.L., 1992, *What Computers Still Can't Do: A Critique of Artificial Reason,* The MIT Press.

38. Dreyfus, H.L. and Dreyfus, S.E., 1986, *Mind Over Machine,* The Free Press, New York.

39. Endsley, M.R., 1997, "The Role of Situation Awareness in Naturalistic Decision Making," *Naturalistic Decision Making*. C.E. Zsambok and G. Klein (Eds.), Lawrence Erlbaum Assoc., New Jersey, Ch. 26, pp. 269-83.

40. Eysenck, M.W., 1993, *Principles of Cognitive Psychology,* Lawrence-Erlbaum.

41. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (Eds.), 1996, *Advances in Knowledge Discovery and Data Mining*, The MIT Press.

42. Ferguson, E., 1993, *Engineering and the Mind's Eye,* 2nd printing, The MIT Press.

43. Fitts, P.M., 1962, "The Functions of Man in Complex Systems," *Aerospace Engineering,* January, 1962.

44. Fitts, P.M. (Ed.), 1951, "Human Engineering for an Effective Air Navigation and Traffic Control System," Washington, D.C.: National Research Council.

45. Flach, J.M., and Bennett, K.B., 1996, "A Theoretical Framework for Representational Design," *Automation and Human Performance: Theory and Applications*, Parasuraman, R. and Mouloua, M. (Eds.), Lawrence Erlbaum Assoc, Ch. 4, pp. 65-87.

46. French, S., 1993, *Decision Theory: An Introduction to the Mathematics of Rationality,* Ellis Horwood Limited, England.

47. Friedman, J.H., 1987, "Exploratory Projection Pursuit," *Journal of the American Statistical Association,* Vol. 82, No. 397, pp. 249-66.

48. Furnas, G.W. and Buja, A., 1994, "Prosection Views: Dimensional Inference Through Sections and Projections," *Journal of Computational and Graphical Statistics,* Vol. 3, No. 4, pp. 323-53.

49. Gill, K.S. (Ed.), 1996, *Human Machine Symbiosis: The Foundations of Human-Centred Design,* Springer-Verlag, New York.

50. Goel, V., 1992, " 'Ill-structured Representations' for Ill-Structured Problems," *Proceedings of the Fourteenth Annual Converence of the Cognitive Science Society,* Vol. 14, pp. 130-35.

51. Golden, M., Siemens, R., and Ferguson, J., 1986, "What's Wrong With Rules?," *Proceedings WEXTEX-86. IEEE Western Conference on Knowledge-Based Engineering and Expert Systems. IEEE ,* pp. 162-5.

52. Gorry, G.A. and Morton, M.S.S., 1971, "A Framework for Management and Information Systems," *Sloan Management Review,* Vol. 13, No. 1, pp. 55-70.

53. Hammond, K.R. Hamm, R.M. Grassia, J. and Pearson, T., 1987, "Direct Comparison of the Efficacy of Intuitive and Analytical Cognition in Expert Judgment," *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 17, No. 5, pp. 753-70.

54. Hancock, P.A. and Chignell, M.H., 1993, "Adaptive Function Allocation by Intelligent Interfaces," ACM, *Intelligent User Interfaces '93,* pp. 227-29.

55. Hancock, P.A. and Scallen, S.F., 1996, "The Future of Function Allocation," *Erogonomics in Design,* Vol. 4, No. 4, pp. 24-29.

56. Hazelrigg, G.A., 1996, "The Implications of Arrow's Impossibility Theorem on Approaches to Optimal Engineering Design," *Journal of Mechanical Design,* Vol. 118, pp. 161-64, June, 1996.

57. Hendler, J., 1989, “Editorial: On the Need for Hybrid Systems,” *Connection Science*, Vol. 1, No. 3, pp. 227-29.

58. Hillier, F.S. and Lieberman, G.J., 1995, *Introduction to Operations Research,* 6th ed., McGraw-Hill, New York.

59. Hoffman, R., 1988, "The Problem of Extracting the Knowledge of Experts from the Perspective of Experimental Psychology,” *Proceedings - 2nd International Symposium on Artificial Intelligence and Expert Systems ,* AMK Berlin, pp. 215-29.

60. Holz, H., and Mosler, K., 1994, "An Interactive Decision Procedure with Multiple Attributes Under Risk,” *Annals of Operations Research,* Vol. 52, pp. 151-70.

61. Hughes, D. and Dornheim, M., 1995, "Accidents Direct Focus on Cockpit Automation,” *Aviation Week and Space Technology,* Jan. 30, 1995.

62. Jacquet-Lagreze, E., and Shakun, M.F., 1982, "Decision Support Systems for Semi-Structured Buying Decisions,” *European Journal of Operational Research,* Vol. 16, pp. 48-58.

63. Jahoda, M., 1989, "Artificial Intelligence: An Outsider's Perspective, *Computers in the Human Context*, Forrester, T. (Ed.), pp. 144-56.

64. Jeang, A., and Flkenburg, D.R., 1995, "Interactive Multiple Criteria Decision Making for Product Development,” *Proceedings of the 3rd International Conference, Computer Integrated Manufacturing,* pp. 253-60.

65. Johhannessen, K., 1988, "Rule Following and Tacit Knowledge,” *Artificial Intelligence and Society,* Vol. 2, No. 4, pp. 287-302.

66. Jones, M.R., 1991, "Interactive Modeling in Decision Support Systems,” *Interacting with Computers,* Vol. 3, No. 2, pp. 167-86.

67. Jordan, N., 1963, "The Allocation of Functions Between Man and Machines in Automated Systems,” *Journal of Applied Psychology,* Vol. 47, pp. 161-165.

68. Kahney, H., 1993, *Prolem Solving: Currrent Issues,* 2nd ed., Open University Press.

69. Kantowitz, B. and Sorkin, R., 1987, "Allocation of Functions,” *Handbook of Human Factors,* G. Salvendy (Ed.), Wiley, New York, pp. 356-69.

70. Keen, P.G.W., 1987, "Decision Support Systems: The Next Decade,” *Decision Support Systems,* Vol. 3, pp. 253-65.

71. Keeney, R.L. and Raiffa, H., 1993, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs,* Cambridge University Press.

72. Klein, G., 1998, *Sources of Power: How People Make Decisions,* The MIT Press.

73. Klein, G.A., and Calderwood, R., 1991, "Decision Models: Some Lessons from the Field,” *IEEE Tranactions on Systems, Man, and Cybernetics,* Vol. 21, No. 5, pp. 1018-1026.

74. Korhonen, P., 1992, "Multiple Criteria Decision Support: The State of Research and Future Directions,” *Computers and Operations Research,* Vol. 19, No. 7, pp. 549-51.

75. Korhonen, P., Moskowitz, H., and Wallenius, J., 1990, "Choice Behavior in Interactive Multiple-Criteria Decision Making,” *Annals of Operations Research,* Vol. 23, pp. 161-79.

76. Koschat, M.A., and Swayne, D.F., 1996, "Interactive Graphical Methods in the Analysis of Customer Panel Data," *Journal of Business and Economic Statistics,* Vol. 14, No. 1, pp. 113-26.

77. Koschmann, T.; Kelson, A.C., Feltovich, P.J.; and Barrows, H.S., 1996, "Computer-Supported Problem-Based Learning: A Principled Approach to the Use of Computers in Collaborative Learning," *CSCL: Theory and Practice of an Emerging Paradigm*, T. Koschmann (Ed.), Ch. 4.

78. Kuchar, J.K. and Hansman, R.J., 1995, *A Unified Methodology for the Evaluation of Hazard Alerting Systems,* MIT Aeronautical Systems Laboratory, ASL-95-1.

79. Kuhn, T.S. , 1996, *The Structure of Scientific Revolutions,* 3rd ed., The University of Chicago Press.

80. Kurzweil, R., 1999, *The Age of Spiritual Machines,* Viking, New York.

81. Langton, C.G. (Ed.), 1987, *Artificial Life,* The proceedings of an interdisciplinary workshop on the synthesis and simulation of living systems held September 1987 in Los Alamos, NM, Addison-Wesley.

82. Larichev, O.I., and Petrovsky, A.B., 1988, "Decision Support Systems for Ill-Structured Problems: Requirements and Constraints," *Organizational Decision Support Systems. Proceedings of the IFIP WG 8.3 Working Conference,* 247-57.

83. Lawrence, J., 1994, *Introduction to Neural Networks: Design, Theory, and Applications,* 6th ed., California Scientific Software

84. Leveson, N.G., 1995, *Safeware: System Safety and Computers,* Addison-Wesley.

85. Levis, A.H., Moray, N., and Hu, B., 1993, "Task Decomposition and Allocation Problems and Discrete Event Systems," *Automatica,* 3(2), 203-16.

86. Lowgren and Stolterman, E., 1999, "Design Methodology and Design Practice," *Interactions of the Association of Computing Machinery,* Jan-Feb 1999.

87. Maes, P., 1995, "Modeling Adaptive Autonomous Agents," *Artificial Life: An Overview,* C.G. Langton (Ed.), MIT Press, pp. 135-61.

88. Manningham, D., 1997, "Where is Automation Going?," *Business and Commercial Aviation,* Sept. 1997.

89. McDonald, J.A., 1982, "Interactive Graphics for Data Analysis," Ph.D. Thesis, Stanford Univ.

90. Medsker, L.R., 1994, *Hybrid Neural Network and Expert Systems,* Kluwer Academic Publishers, Massachusetts.

91. Meister, D., 1971, *Human Factors: Theory and Practice,* Wiley, New York.

92. Mesavoric, M.D., 1970, *Theory of Hierarchical, Multilevel Systems,* Academic Press, New York.

93. Miller, G.A., 1956, "The Magic Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information," *The Psychological Review,* Vol. 63, No. 2, pp. 81-97.

94. Minns, A.W., 1998, *Artificial Neural Networks as Subsymbolic Process Descriptors*, A.A. Balkema, Vermont.

95. Minsky, M., 1994, "A Conversation with Marvin Minsky About Agents," *Communications of the ACM,* Vol. 37, No. 7, 23-29, July, 1994.

96. Minsky, M., 1986, *The Society of Mind,* Simon and Schuster, New York.

97. Minsky, M., , 1961, "Steps Toward Artificial Intelligence," *Proceedings of the IRE,* Vol. 49, pp. 8-29.

98. Mitchell, T.R., and Beach, L.R., 1990, " 'Do I Love Thee? Let Me Count' Toward an Understanding of Intuitive and Automatic Decision Making," *Organizational Behavior and Human Decision Processes,* Vol. 47, pp. 1-20.

99. Mond, B. and Rosinger, E.E., 1985, "Interactive Weight Assessment in Multiple Attribute Decision Making," *European Journal of Operational Research,* Vol. 22, pp. 19-25.

100. Monsay, E.H., 1997, "Intuition in the Development of Scientific Theory and Practice," *Intuition: The Inside Story*, R. Davis-Floyd and P.S. Arvidson (Eds.), Routledge, New York, Ch. 6, pp. 103-20.

101. Mosier, K.L. and Skitka, L.J., 1996, "Human Decision Makers and Automated Decision Aids: Made for Each Other?," *Automation and Human Performance: Theory and Applications*, R. Parasuraman and M. Mouloua, (Eds.), Lawrence Erlbaum, Ch. 10, pp. 201-219.

102. Newell, A. and Simon, H., 1976 "Computer Science as Empirical Inquiry", Reprinted in *Mind Design*, J. Haugeland (Ed.), The MIT Press, pp. 35-66.

103. Norman, D.A., 1992, "Design Principles for Cognitive Artifacts," *Research in Engineering Design,* Vol. 4, pp. 43-50.

104. Norman, D.A., 1989, "The 'Problem' of Automation: Inappropriate Feedback and Interaction, Not 'Overautomation' ," Prepared for the discussion meeting, *Human Factors in High-Risk Situations, The Royal Society* (Great Britain), June 28-29, 1989.

105. Norman, D.A., 1988, *The Psychology of Everyday Things,* Basic Books, New York.

106. Norman, D.A., 1993, *Things That Make us Smart,* Addison-Wesley, New York.

107. Norman, D.A., 1992, *Turn Signals are the Facial Expressions of Automobiles,* Addison-Wesley, New York.

108. Norman, D.A., 1997, "Why It's Good That Computers Don't Work Like The Brain," *Beyond Calculation: The Next Fifty Years of Computers,* P. Denning and R. Metcalfe (Eds.), Springer-Verlag, New York, Ch. 8, pp. 163-82.

109. Orasanu, J., and Connolly, T., 1993, "The Reinvention of Decision Making," *Decision Making in Action: Models and Methods*, G.A. Klein; J. Orasanu; R. Calderwood; and E. Zsambok (Eds.), Ch. 1, pp. 3-20.

110. Otto, K.N., 1993, "Measurement Foundations for Design Engineering Methods," *ASME Design Theory and Methodology 1993*, DE-Vol. 53, pp. 157-66.

111. Otto, K.N., and Antonsson, E.K., 1993, "The Method of Imprecision Compared to Utility Theory for Design Selection Problems," *ASME Design Theory and Methodology 1993*, DE-Vol. 53, pp. 167-73.

112. Patrick, N.J.M., 1996, "Decision-Aiding and Optimization for Vertical Navigation of Long-Haul Aircraft," Ph.D. Thesis, MIT.

113. Pahng, F., Senin, N., and Wallace, D., 1998, "Distributed Object-Based Modeling and Evaluation of Design Problems," MIT Computer Aided Design Laboratory Publication, Dept. of Mechanical Engineering, URL:http://cadlab.mit.edu/publications/ [cited May 21, 1999].

114. Polyani, M., 1967, *The Tacit Dimension,* Anchor Books, Doubleday and Company, New York.

115. Pomerol, J., 1993, "From Aggregating by Rules to the Integration of Expert Systems in Multicriteria Decision Support Systems," *1993 International Conference on Systems, Man and Cybernetics*, Systems Engineering in the Service of Humans, New York, Vol. 1, pp. 483-8.

116. Price, H.E., 1985, "The Allocation of Functions in Systems," *Human Factors,* Vol. 27, No. 1, pp. 33-45.

117. Pritchett, A.R., and Hansman, R.J., 1997, "Pilot Non-Conformance to Alerting System Commands During Closely Spaced Parallel Approaches," Aeronautical Systems Laboratory, MIT, ASL-97-2.

118. Pritchett, A.R., Hansman, R.J., and Johnson, E.N., 1996, "Use of Testable Responses for Performance-Based Measurement of Situation Awareness," Presented at the International Conference on Experimental Analysis and Measurement of Situation Awareness, Daytona Beach, FL, November, 1996.

119. Ramaswamy, R. and Ulrich, K., 1993, "A Designer's Spreadsheet," *ASME Design Theory and Methodology 1993*, DE-Vol. 53, pp. 105-13.

120. Rasmussen, J., 1986, *Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering,* Elsevier Science Publishing Co., New York.

121. Reason, J., 1988, "Cognitive Aids in Process Environments: Prostheses or Tools?," *Cognitive Engineering in Complex Dynamic Worlds,* E. Hollnagel, G. Mancini, and D.W. Woods (Eds.), Academic Press, London, Ch. 1, pp. 7-15.

122. Reitman, W.R., 1965, *Cognition and Thought: An Information-Processing Approach,* John Wiley and Sons, New York.

123. Reitman, W.R., 1964, "Heuristic Decision Procedures, Open Constraints, and the Structure of Ill-Defined Problems," *Human Judgments and Optimality,* M.W. Shelly and G.L. Bryan (Eds.), Ch. 15, pp. 282-315.

124. Ringuest, J.L., 1992, "Implementing Multiobjective Optimization Methods: Behavioural and Computational Issues," *Computers and Operations Research,* Vol. 19, No. 7, pp. 547-48.

125. Roseborough, J.B., 1988, "Aiding Human Operators with State Estimates," Ph.D. Thesis, MIT.

126. Rosenhead, J., 1989, *Rational Analysis for a Problematic World,* J. Rosenhead (Ed.), John Wiley and Sons Ltd., England, Ch. 1.

127. Rouse, W.B., 1991, *Design for Success: A Human-Centered Approach to Designing Successful Products and Systems,* John Wiley and Sons, New York.

128. Rouse, W.B., and Hammer, J.M., 1991, "Assessing the Impact of Modeling Limits on Intelligent Systems," *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 21, No. 6, pp. 1549-59.

129. Rouse, W.B. and Cody, W.J., 1986, "Function Allocation in Manned System Design," in Proc: *1986 IEEE Int. Conf. Systems, Man, and Cybernetics*, IEEE, New York, pp. 1600-1606.

130. Sage, A.P., 1981, "Behavioral and Organizational Considerations in the Design of Information Systems and Processes for Planning and Decision Support," *IEEE Tranactions on Systems, Man, and Cybernetics,* Vol. 11, No. 9, pp. 640-78.

131. Schultz, R., 1994, *Unconventional Wisdom,* HarperCollins, Ch. 1.

132. Scriabin, M., Kotak, D.B., Whale, K.G., 1995, "Symbiotic Systems: Exploiting Human Creativity," *European Journal of Operational Research,* Vol. 84, No. 2, pp. 227-34.

133. Shelly, M.W., and Bryan, G.L., 1964, "Judgments and the Language of Decisions," *Human Judgments and Optimality*, M.W. Shelly and G.L. Bryan (Eds.), JohnWiley and Sons, Inc., New York, Ch.1, pp. 3-37.

134. Shepard, R.N., 1964, "On Subjectively Optimum Selection Among Multiattribute Alternatives," *Human Judgments and Optimality*, M.W. Shelly and G.L. Bryan (Eds.), Ch. 14, pp. 257-81.

135. Sheridan, T.B., 1996, "Automation and Human Performance: Looking Ahead into the 21st Century," *Human-Automation Interaction*, M. Mouloua, and J.M. Koonce (Eds.), Lawrence Erlbaum.

136. Sheridan, T.B., 1995, "Human Centered Automation: Oxymoron or Common Sense?," Keynote address, IEEE Systems, Man, and Cybernetics Conference, Vancouver, BC, Oct. 23-25, 1995.

137. Sheridan, T.B., 1996, "Speculation on Future Relations Between Humans and Machines, *Automation and Human performance: Theory and applications*, R. Parasuraman and M. Mouloua (Eds.), Lawrence Erlbaum, Ch. 21, pp. 449-60.

138. Sheridan, T.B., 1992, *Telerobotics, Automation, and Human Supervisory Control,* The MIT Press.

139. Sheridan, T.B., 1998, "Allocating Functions Rationally Between Humans and Machines," *Ergonomics in Design,* Jul-98.

140. Simon, H.A., 1997, "The Future of Information Systems," *Annals of Operations Research,* Vol. 71, pp. 3-14.

141. Simon, H.A., 1996, *The Sciences of the Artificial,* 3rd ed., Cambridge, MA: MIT Press.

142. Simon, H.A., 1973, "The Structure of Ill Structured Problems," *Artificial Intelligence,* Vol. 4, pp. 181-201.

143. Simon, H.A., 1987, "Two Heads are Better than One: The Collaboration between AI and OR," *Interfaces,* Vol. 17, No. 4, pp. 8-15.

144. Simon, H.A., Dantzig, G.B., Hogarth, R., Plott, C.R., Raiffa, H., Schelling, T.C., Shepsle, K.A., Thaler, R., Tversky, A., and Winter, S., 1987, "Decision Making and Problem Solving," *Interfaces,* Vol. 17, No. 5, pp. 11-31.

145. Simon, H.A., 1977, *The New Science of Management Decision*, Prentice Hall, Englewood Cliffs.

146. Suh, N.P., 1990, *The Principles of Design,* Oxford University Press, New York.

147. Taylor, J.G., 1996, *Neural Networks and Their Applications,* John Wiley and Sons, New York.

148. Thurston, D.L., 1993, "Subjective Design Evaluation with Multiple Attributes," *ASME Design Theory and Methodology 1993*, DE-Vol. 53, pp. 355-61.

149. Travers, M.D., 1996, "Programming with Agents: New Metaphors for Thinking About Computation," Ph.D. Thesis, MIT.

150. Truxal, J.G., 1961, "The Concept of Adaptive Control," *Adaptive Control Systems,* E. Mishkin and L. Braun (Eds.), Ch. 1, pp. 1-19.

151. Vessey, I., 1994, "The Effect of Information Presentation on Decision Making: A Cost-Benefit Analysis," *Information and Management,* Vol. 27, pp. 103-119.

152. Vicente, K.J., and Rasmussen, J., 1992, "Ecological Interface Design: Theoretical Foundations," *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 22, No. 4, pp. 589-606.

153. Von Neuman, J. and Morgenstern, O., 1947, *Theory of Games and Economic Behaviour,* Princeton University Press.

154. Wallace, D.R. Jakiela, M. and Flowers, W., 1995, "Design Search Under Probabilistic Specifications Using Genetic Algorithms," *Computer-Aided Design*, June 1995.

155. Wallace, D.R. and Jakiela, M., 1993, "A Computer Model of Aesthetic Product Design: An Approach to Unify Engineering and Industrial Design," MIT Computer Aided Design Laboratory Publication, Dept. of Mechanical Engineering, URL:http://cadlab.mit.edu/publications/ [cited May 21, 1999].

156. Waltz, D.L., 1997, "Artificial Intelligence: Realizing the Ultimate Promises of Computing," *AI Magazine,* pp. 49-51, Fall, 1997.

157. Wang, P.C.C., 1978, *Graphical Representation of Multivariate Data,* Academic Press.

158. Weizenbaum, J., 1976, *Computer Power and Human Reason,* W.H. Freeman and Co., New York.

159. White, D. and Sofge, D. (Eds.), 1992, *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches,* Multiscience Press, New York.

160. Wiener, 1964, *God and Golem, Inc.,* The MIT Press.

161. Wiener, N., 1961, *Cybernetics,* 2nd ed., The MIT Press.

162. Winograd, T., 1995, "Thinking Machines: Can There Be? Are We?," *Informatica,* Vol. 19, pp. 443-459.

163. Winston, P., 1997, "Rethinking Artificial Intelligence," URL: http://www.ai.mit.edu/director/briefing.html [cited 23 July, 1998].

164. Winterfeldt, D.V., 1980, "Structuring Decision Problems for Decision Analysis," *Acta Psychologica,* Vol. 45, pp. 71-93.

165. Yntema, D.B., and Torenson, W.S., 1961, "Man-Computer Cooperation in Decisions Requiring Common Sense," *IRE Transactions on Human Factors in Electronics,* HFE-2, pp. 20-26.

166. Yolles, M., 1998, "Changing Paradigms in Operational Research," *Cybernetics and Systems: An International Journal,* Vol. 29, pp. 91-112.

167. Yu, P.L., 1992, "To Be a Great Operations Researcher from a MCDM Scholar," *Computers and Operations Research,* Vol. 19, No. 7, pp. 559-61.

168. Zeleny, M., 1992, "An Essay into a Philosophy of MCDM: A Way of Thinking or Another Algorithm?," *Computers and Operations Research,* Vol. 19, No. 7, pp. 563-66.

169. Zhao, S. and Shen, S., 1989, "The Completeness Problem of Knowledge Bases," *Proceedings of the Sixth IASTED International Symposium. Expert Systems: Theory and Application*, Los Angeles, CA., 14-15 Dec. 1989.