Semi-Structured Decision Processes

William N. Kaliardos

Professor R. John Hansman
Massachusetts Institute of Technology

Introduction

Determining the appropriate role of automation within a human-automation system is a key design task. “Function allocation” is the task of allocating work between humans and computers. In order to improve function allocation, a simple model of decision processes—known as “Semi-Structured” processes—is presented as the basis of a conceptual framework for understanding decision systems.

Allocating Functions Between Humans and Computers

A decision system is a set of components that is designed to satisfy a set of system functions during the period of “operation.” These components are typically realized with humans and automation. Example systems include aircraft cockpits, power plant control, and decision support systems in management and medicine.

One of the difficulties associated with allocating functions between humans and computers is that there lacks a single metric for utility [12]. In the absence of prescriptive design methodologies ([4], [5], [7], [9]), function allocation is often performed *ad hoc*. This research attempts to provide insight into decision systems, without prescribing a methodology for their design.

Semi-Structured Processes

A decision process is defined as “Structured” when it can be reduced to well-defined rules. Conversely, an “Unstructured” process is one that cannot be reduced to well-defined rules. A process that contains both Stuctured and Unstructured elements is defined here as a “Semi-Structured” process:

> *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 believed that many decisions use processes that are Semi-Structured. 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 can be complex and ill-defined. Semi-Structured decision processes are also observed in many other human-automation systems, as well as in fully-human and fully-automated systems.

The notion of Semi-Structure is applied to processes—actions taken to arrive at a decision output—and not to their intended functions (as in [14]). Simply put, Structure applies to the "hows" and not the "whats." Generally, there are many processes that can satisfy a function ([2], [8]) and hence, many ways to use humans and automation within a system. One approach is to use a Structured process for decision making. This has typically been the strategy within the field of Artificial Intelligence (e.g., Expert Systems). However, it can be inappropriate to impose excessive Structure on a decision process, in which case Unstructured processes may be valuable components.

The Structured Process

Elaborating on the previous definitions, a "rule" is a special type of process, since it is a process that can be *represented*. Furthermore, a rule is "well-defined" when the process can be *unambiguously represented*—for example, in the language of formal logic or mathematics.

An important implication of Structure is that the process is a symbolic process that is prescribed prior to its operation, with assumptions about the environment in which the symbols will operate (via inputs and outputs). Structured processes therefore tend to be rigid: performing the same input-output transformation independent of the operational environment. Just as a word may have an intended meaning only in certain contexts, a symbolic process may satisfy a function only in certain environments. Hence, inappropriate assumptions made during *design* can lead to unanticipated behavior during *operation* ([13], [16]).

Lastly, while it is not necessary for a Structured process to be articulated in computer code, or realized on machines, computer code is a sufficient condition for Structure:

> *A <u>test</u> for Structure is when a process can be reduced to a traditional computer algorithm (e.g., not self-organizing)*

The Unstructured Process

Since an Unstructured process cannot be reduced to well-defined rules, it can be thought of as a "black box". It may be possible to define inputs and outputs, but the underlying process is not decomposable to unambiguous rules. Some important implications of Unstructured processes are:

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 to a decision 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.

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.

Another way of looking at Unstructured processes is that they are left open during the *design* stage, to be filled in at a later date: during *operation*. Such processes may help the system handle complexity during operation.

Reasons Why Structure May Not Be Appropriate

There are many reasons why it may not be appropriate to fully Structure a decision process. These reasons are categorized in Figure 1 (a thorough review of these are provided in [6]). For example, one implication of rules is that they require a minimum or sufficient set of inputs, which becomes problematic if this information is unavailable. The remainder of this paper focuses on *complexity* (Category *A*).

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

Figure 1 Categorized reasons why Structure may be inappropriate

Accommodating Complexity

A Structured process may not be appropriate due to complexity, which inherently implies that a decision situation is not well understood in an explicit sense. For example, medical diagnosis can involve large numbers of potentially relevant indicators—from gene markers to macroscopic symptoms—but the fusion of this information via algorithms is limited due to the complexity of human physiology. In such cases, we often find that Unstructured processes are valuable decision components. Furthermore, Structure is often used as a reliable way to *manage* complexity within a Semi-Structured process.

Advantages of Unstructured Processes

Unstructured processes might be able to accommodate complexity because decision making does not have to be based on an explicit understanding. This appears to be have implications on both humans and computers.

Humans are known for their ability to handle complexity. It is believed that, with experience, humans are able to reduce *patterns* of otherwise complex information into simple "chunks" that can be easily recognized [13]. In particular, 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 appears to be one reason why experts are less likely to fall victim to information overload. Humans also use techniques such as heuristics and abstraction hierarchies to circumvent complexity.

As an example of *automated* Unstructured processes, neural networks are also known for their ability to generalize correctly from complex data sets [15]. They may be trained without the need for humans to ever *explicitly* understand the process. This can lead to the discovery of data patterns, which is particularly advantageous in massive-data situations, where perception cannot be exploited and/or short-term memory becomes a bottleneck. As with humans, previous exposure to the right type of training scenarios is critical [10].

Humans and Automation

The abstraction of a decision process as Semi-Structured does not imply a specific allocation strategy. For example, Structure does not *imply* automation, but it may *suggest* automation. Similarly, Unstructure does not imply, but suggests allocation to humans. Allocation decisions can involve many complex issues beyond the scope of this paper, and these decisions are ultimately left to the judgment of the designers of a specific system. The purpose of this section is to understand the realization of Structured and Unstructured components with humans and automation.

Realizing Structured Processes

It may be clear that a Structured process can often be realized on machines such as digital computers. Traditionally, Structure is often a necessary condition for automating. Hence, if it is desired to automate a certain function, a process must first be articulated in the form of well-defined rules. The resulting automated process can then likely be executed with greater precision, repeatability, and speed than with a human. However, it is the inherent property of the Structured process that provides such properties ; automation merely provides advantages in execution.

Human-executed Structure provides a means for designing and organizing human behavior. Some benefits of Structure are that it allows:

- multiple people to share their collective wisdom with an explicit representation
- novices to make good decisions, even without an understanding of the underlying reasoning

Standard operating procedures, for instance, allow pilots to perform fast but good decisions during time-critical emergencies. Since they are explicitly represented, such instructions can be modified to evolve over time. Whether executed with humans or automation, a Structured process needs to be applied in the appropriate context—as determined during design.

Realizing Unstructured Processes

Unstructured processes *suggest* a need for humans. It appears that humans add value to Unstructured processes. In addition, it is argued that certain self-organizing or emergent algorithms can also be considered Unstructured.

The natural way of human decision making is believed to be Unstructured. That is, most decisions involve judgments and intuition, without conscious analysis of the underlying logic. Even with sufficient analysis it can be difficult to understand human decision making in the form of rules. For example, the development of Expert Systems appears limited by the knowledge that can be elicited from human experts.

It is reasonable to also consider certain *automated* processes as Unstructured. The justification for this is that, although there is no argument that the core behavior stems from well-defined rules (i.e., machine code), the behavior of the process during operation is not one that was explicitly prescribed during design. For example, neural networks are self-organizing processes that often learn by example, much the way humans can learn. The result is that the operational neural network decision process behaves in a way that is not predictable from its original rule set, and may not even be understandable if the operational logic is exposed.

Diagrammatic Notation

Figure 2 introduces a diagrammatic notation for representing decision systems as Semi-Structured processes. This notation is valuable for organizing the concepts discussed earlier. In particular, it provides a way to represent the ill-defined components that are generally associated with Unstructured processes. By representing systems in this manner, designs can be considered and compared more explicitly as a basis for understanding implications of function allocation strategies.

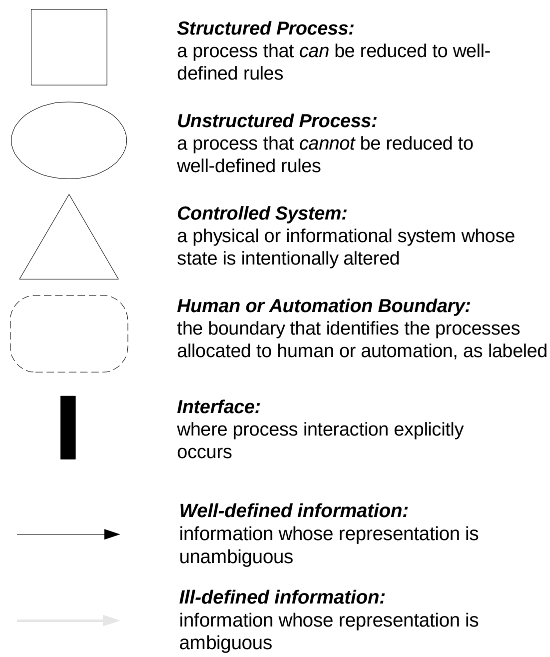

Figure 2 Diagrammatic notation for Semi-Structured Decision Processes

Example: Aircraft Cockpit Decision Making

The following example illustrates how Structure is used within a Semi-Structured process to manage complexity. Aircraft cockpits are assumed to represent good designs, based on the fact that they have evolved this way over considerable time. In these systems, the human remains a functional decision component—modeled here as an Unstructured process—and a key element to the success of these designs has been the ability to deal with complexity during flight.

Automated *controllers* are one of the most common examples of Structured decision making, and the first type of automation used in aircraft. Figure 3 illustrates an aircraft cockpit decision process with multiple control modes. Each controller is similar in that it takes a well-defined input (goal or target state), and provides decision outputs—perhaps based on optimal control laws. These processes hide complexity from the human. However, a consequence is that interfaces are required to communicate relevant information. One advantage is that this information provides good situation awareness for the human supervisor [3], who is particularly valuable for responsibility and adaptivity outside of nominal flight conditions [1].

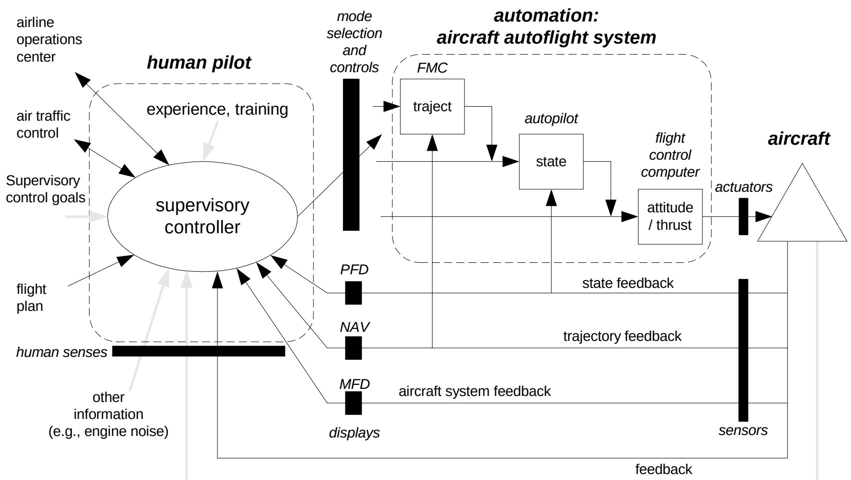

Figure 3 Aircraft cockpit: a Semi-Structured decision process

In addition to *control* automation, *information* automation also hides complexity—using Structure to process information "upstream" from the Unstructured process (not shown in Figure 3 above). Information automation is intended to empower the pilot by preventing information overload, and presenting the appropriate information at the right time and in an appropriate format. As with automated controllers, these automated "observers" lie on a spectrum of automation, as shown in Figure 4.

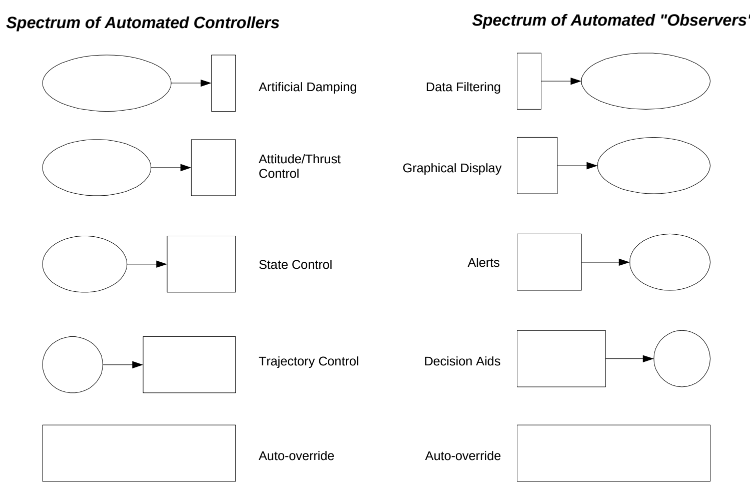

Figure 4 Various uses of Structure in aircraft automation

Conclusions

This paper introduced the Semi-Structured process as a conceptual framework for understanding decision systems—particularly for providing insight as to where automation makes sense, and where humans add value. For example, Unstructured processes tend to move a system away from optimality during nominal conditions, but tend to add flexibility and robustness during non-nominal situations. In particular, the framework helps make the ill-defined components of a decision system explicit, such that automation is used appropriately. This is evident in the control of complex systems such as modern aircraft, where automation hides complexity, and empowers the pilot by helping to make decisions more informed.

References

1. Billings, C.E., 1996, *Human-Centered Aviation Automation: Principles and Guidelines,* NASA Technical Memorandum 110381.
2. Blanchard, B.S. and Fabrycky, W.J., 1981, *Systems Engineering and Analysis,* Prentice-Hall, New Jersey, pp. 2-15.
3. 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.
4. Fitts, P.M. (Ed.), 1951, "Human Engineering for an Effective Air Navigation and Traffic Control System," Washington, D.C.: National Research Council.
5. Hancock, P.A. and Scallen, S.F., 1996, "The Future of Function Allocation," *Erogonomics in Design,* Vol. 4, No. 4, pp. 24-29.
6. Kaliardos, W., 1999, "Semi-Structured Decision Processes: A Conceptual Framework for Understanding Human-Automation Systems," Ph.D. Thesis, MIT.
7. Kantowitz, B. and Sorkin, R., 1987, "Allocation of Functions," *Handbook of Human Factors,* G. Salvendy (Ed.), Wiley, New York, pp. 356-69.
8. Lowgren and Stolterman, E., 1999, "Design Methodology and Design Practice," *Interactions of the Association of Computing Machinery,* Jan-Feb 1999.
9. Price, H.E., 1985, "The Allocation of Functions in Systems," *Human Factors,* Vol. 27, No. 1, pp. 33-45.
10. 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.
11. Sheridan, T.B., 1992, *Telerobotics, Automation, and Human Supervisory Control,* The MIT Press.
12. Sheridan, T.B., 1998, "Allocating Functions Rationally Between Humans and Machines," *Ergonomics in Design,* Jul-98.
13. Simon, H.A., 1996, *The Sciences of the Artificial,* 3rd ed., Cambridge, MA: MIT Press.
14. Simon, H.A., 1973, "The Structure of Ill Structured Problems," *Artificial Intelligence,* Vol. 4, pp. 181-201.
15. Taylor, J.G., 1996, *Neural Networks and Their Applications,* John Wiley and Sons, New York.
16. Wiener, 1964, *God and Golem, Inc.,* The MIT Press.