On Information and Performance of Complex Manufacturing Systems

Albert T. Jones and Larry H. Reeker

Manufacturing Engineering Laboratory
National Institute of Standards and Technology
Gaithersburg, MD

Abhijit V. Deshmukh

Mechanical & Industrial Engineering
University of Massachusetts
Amherst, MA

ABSTRACT

This paper makes several critical points related to the intrinsic relationship between information and performance in complex, dynamic systems. Using a manufacturing enterprise as an example, we highlight the co-dependence of information and decision outcomes in such systems, stress the impact of this co-dependence on the evolution and performance of these systems, and, raise research issues that are fundamental to a better understanding of both. We claim that the availability of the right information at the right time is crucial for making good decisions. We argue further that determining whether available information is indeed the right information is a difficult problem. Determining the difference between two information objects, and moreover, the impact of this difference on the decisions made needs to be investigated. Finally, we discuss the dependence of the value of an information object and the decision being made using that object.

INTRODUCTION

Manufacturing is changing in two important ways. At the shop floor level, automated data collection systems support the movement toward real-time scheduling decisions. At the enterprise level, the Internet supports the movement toward global, supply-chain-management decisions. In both cases, the availability of required information at the appropriate time is crucial for making good decisions. Current technology can assure the availability of information, but it cannot assure that the information is accurate and meaningful. Many decisions are made by optimization algorithms implemented in software applications. These applications assume that required inputs are either stored in a local repository or can be acquired from other sources in the enterprise when and as needed. Furthermore, they assume that these inputs are current, accurate, and meaningful. Of the three, the third is, by far, the most difficult because it requires a determination of semantic differences between information objects. Consider two simple examples. First, on the shop floor, knowing that all applications mean the same thing by the term *duration* is crucial to making good scheduling decisions. Second, in emergent enterprise structures such as supply chains where the participants are constantly changing, knowing that they all mean the same thing by the term *due date* is crucial to making good planning decisions. Since these, and many other, decisions have a direct impact on the performance of the enterprise, we argue that information and performance are intrinsically linked. How to quantify this causal relationship between information and performance is still an open research question.

Information needed to make decisions must be conveyed in physical symbols like marks on paper, sounds, electrical pulses, etc. Nevertheless, information has an effect on a system that is not explainable by its physical properties alone. That effect is related to (1) the *organization* of the symbols, (2) the *meaning* ascribed to the symbols and their organization, and (3) the change in the system *state* that comes from understanding and acting on that meaning. Since the state varies with time, so will the effect of a particular item of information on the properties of the system.

In the earliest applications, the physical carriers of information were, for example, mechanical links in steam engine governors, punched holes in Jacquard looms, or electrical connectivity in thermostats. The impacts of the information could be measured for these applications by its physical properties, cause, and effect -- the increase in heat, the movement of the engine or loom, the current flow in a thermocouple, and so on. Since those outcomes could be measured and quantified, the impact of the information on the system performance could be observed. The *meaning* of the information, not its *representation*, was what influenced that performance. The punched holes in the loom cards were originally in stiff pasteboard and were read by needles. After their evolution into Hollerith's paper cards, they could be read by pins that conveyed electricity and later by light and electricity. Finally, when the cards went away entirely in favor of other information representations, the same information could be conveyed by different physical means. Thus, the performance of many physical systems is, in some sense, independent of the physical form of the information that drives them.

The complexity of systems has evolved considerably over the past twenty years. During the past few years increasing global competition has made it essential for manufacturing firms to enhance productivity, shorten product life cycles, increase product customization, and improve responsiveness to remain viable in the market. In order to achieve these seemingly conflicting objectives, the trend in manufacturing systems has been towards integrating all manufacturing functions. Such integrated systems, also referred to as *agile*, *lean*, or *flexible* depending on their level of integration, possess one common characteristic, namely the ability to react to disturbances or changes. These disturbances could occur due to the introduction of new products, volatile consumer demand, changing management objectives, or operating uncertainties such as machine failures or variable processing times. Successful implementation of these integrated systems is contingent on the *correct* reaction being taken to any disturbance. However, due to an incomplete understanding of the interactions among system components and the inability to exchange meaningful information between components, it has been difficult to (1) make decisions about the design of these systems and (2) prescribe effective operating strategies once the designs have been implemented. Often manufacturing system are described as *complex* [Pritsker, 90]. In fact, Upton [Upton, 88] observes that many of these integrated manufacturing systems, which are designed to be flexible, are constrained by their decision complexity to be inflexible. On the manufacturing shop floor, especially one with dynamic controls and an unpredictable environment, decisions are constantly being made concerning the system. Each decision contributes to aggregate system performance. Therefore, an understanding of the effects of the impact of the informational component, its representation and its meaning, is paramount. It is clear that testing for the amount of and the impact of information in any particular area is going to be difficult, and that even the terms "amount" and "impact" will be difficult to define. In short, a metric of the information abstracted from the physical parameters is not evident. Moreover, we postulate that the complexity of the decision being made and the information used in that decision are tightly coupled. Hence, the amount and impact of information may not be defined independent of the decisions.

IMPACT OF INFORMATION ON CONTROL OF COMPLEX SYSTEMS

Two of the simple systems mentioned above as examples – the thermostat and the governor – are ones in which the information is gathered by feedback, which is the collection of information, its representation in a physical medium, and its interpretation to control a system. The handling of feedback information for controlling complex systems can be much less direct. Yet it is often the case that simple models can provide ideas that can be generalized to more complex ones, and maybe it can help in this case to understand the general problem of measuring information impact.

Many complex systems can be viewed as a collection of integrated and layered subsystems, which might at some "bottom" layer be cases of direct physical control. Typically, the layering occurs in both the temporal domain and the spatial domain. The bottom layer contains some combination of biological, chemical, and physical processes. In humans, the evolution of these processes is governed by an internal and natural intelligence, which we call the mind. While we do not know exactly how it works, we see its benefits every moment of our lives.

Man-made systems, on the other hand, are not endowed with a mind. The processes that make up these systems are subject to the second law of thermodynamics. Hence, without any external intelligence to guide their evolution, entropy will increase and they will go out of control over time. To keep this from happening, researchers have expended an enormous amount of time, energy, and money to develop models, algorithms, and heuristics that come under the general heading of control theory.

While it is conceptually straightforward, control theory can be complicated in practice. At an abstract level, it consists of two steps. Step 1 has two parts: set the desired goal and develop a plan to achieve that goal. Step 2 also has two parts: observe plan execution and make adjustments as required. Step 1 usually involves the creation of a system model, an optimization problem based on that model, and technique to solve that problem. Models, which combine continuous, discrete, deterministic, and stochastic aspects, typically have temporal and spatial parameters. The optimization problem has at least one measurable goal and constrains the parameters in the model. Sometimes these problems can be solved analytically, sometimes not. Regardless of how the solution is derived, it results in a plan to be executed by the system.

Consider a robot that that must move a part from point A to point B in the shortest possible time. To generate a path to accomplish this goal, the robot controller, which could be a human or a software procedure, needs models of the robot and its environment. These models are continuous time, continuous state, and assumed deterministic. The controller formulates an optimization problem whose solution will specify the start coordinates, the end coordinates, the duration, limits on models parameters (such as speed, joint angles, and so on), and possible obstacles to avoid. That solution yields the optimal plan that the robot should use. This plan is then sent to the robot, or more accurately the execution part of its controller, to be implemented. Once the robot begins to move, we must proceed to step 2. This means that we must somehow make sure that the robot does not exceed any of the limits and follows the predetermined path. We do this through the generation and analysis of feedback. Sensors on the robot create the feedback, which is analyzed by the controller. When a problem is detected, a new plan will be generated.

CRITICAL POINT 1: Both the plan and the feedback are information objects, which impact the performance and the behavior of the robot. Some of these objects are simple; some are not. The meaning of these objects must be conveyed to and understood by all hardware and software components or there is no hope of achieving the desired goals. These capabilities do not happen "naturally"; they must be built into the system.

As we move up the layers, we no longer deal directly with biological, chemical, and physical systems. Instead, we deal with decision-making and information systems that affect those bottom-layers, but on a longer-term basis. Nevertheless, the same two steps are involved. In this case, however, the models are typically discrete time and discrete state systems that often contain both deterministic and stochastic parameters. There are several, often conflicting, quantitative performance measures and the techniques are implemented in a number of software applications such as linear programming, demand forecasting, and supply chain management. These applications also produce plans that are implemented in other, lower-layer software applications -- demands lead to production plans, which lead to schedules, which lead to sequences and so on. These plans are based on information that has a high degree of uncertainty. Some of this uncertainty arises because of the influence of the second law on the bottom-layer processes. Some of it arises because of the stochastic nature of predictions associated with demand projections, priority orders, and material arrivals, to name a few, at higher layers.

CRITICAL POINT 2: Optimizing performance measures is critically dependent on the ability of the associated software applications to share complex information objects. Furthermore, without having a common understanding of the meaning of those objects, optimization is useless.

As we progress through the various layers of a complex system like a manufacturing enterprise, an evolution occurs from continuous time to discrete time and from continuous state to discrete state. Furthermore, an aggregation in information takes place as well - very detailed, relatively simple, deterministic information at the bottom; very little detail, more complex, highly stochastic information at the top. No one knows how this evolution or aggregation takes place. Moreover, at every layer, there is some influence of entropy from both the second law and information uncertainty. At the bottom, the second law dominates. At the top, information uncertainty dominates. We have a very good idea of how to measure and control the effects of the second law on physical system performance. We have almost no idea how to measure and control the effects of information uncertainty on system performance.

CRITICAL POINT 3: Information has a large impact on system performance. Integration, getting the right information from one software application to another, also has an impact. Consequently, ensuring that all software applications have the same understanding of that information is critical to system performance. Furthermore, and most importantly, our ability to measure how well they understand impacts directly our ability to measure the true performance of the system.

An important question then is how can we build software applications that are capable of understanding information and dealing with its inherent uncertainty. The simple answer is that we must make software, just as we must make equipment, more intelligent. More accurately, we must surround each software application with the "stuff" it needs to understand the information it receives from other applications. A partial list includes:

Parsers to determine the structure of an encoding (the physical representation) according to a known structural description (for symbols, called a "grammar").

Ontologies to describe the internal model that the system can use to recognize inputs in terms of catalogues of entities and processes and their relationships.

Dictionaries to define the relationship of discrete elements of the encoding to objects and processes in the ontology.

Mappers from encodings to models or directly from one model to another.

Controller which makes decisions on a course of action (a sequence of behaviors), based on information in plans that have been preprogrammed or formulated and inputs (from users or sensors, including feedback), and operates actuators to cause the behavior sequence.

Actuators: Devices that behave physically to produce behavior.

Perceptors: Systems that convert the input of sensors into information for the system to process.

Equivalence Metrics: Ways of measuring how the information in one system or subsystem relates to another – whether it is equivalent or not.

CRITICAL POINT 4: Our ability to control the performance of the physical systems can depend directly on our ability to measure the similarities and differences between information objects.

MEASURING EQUIVALENCE OF INFORMATION OBJECTS

Perhaps the first thing to consider in looking at measurement metrics is whether definitions of equivalence can be established. This is a tricky issue, because in some sense they cannot. Consider two ontologies, as defined above. They conventionally are represented by classes of entities and their attributes, linked into hierarchies or lattices based on the IS-A relationship. IS-A relationships are based on the attributes of classes of entities, and those attributes are based two things: fundamental properties and the behaviors of entities in activities. Trying to compare behaviors of entities after a certain degree of complexity is reached leads to things like the halting problem. Thus, just as it may be formally undecidable if two programs are equivalent, it may be difficult to assert ontological equivalence formally. Perhaps we can still get measurements that will enable satisfactory performance within bounds, and undecidability will not be a problem. We still need to measure some concept of equivalence, even with the blanket restriction of undecidability, which is a common restriction that must be sidestepped often in computing. One approach is to use approximations, which are often required by limited measurement precision anyway.

CRITICAL POINT 5: The equivalence of information objects may be undecidable, but we may be able to develop approximate measurements.

Developing an approximate equivalence metric puts us right in the middle of an ongoing controversy. That controversy revolves around the best way to represent uncertainty in information. There are two views: probabilistic and fuzzy. The probability proponents argue that there is only one consistent way to measure uncertainty and that is probability theory. They further argue that all probability is conditioned upon prior information and that the proper way to do inferencing must be based on a Bayesian framework. That framework says (1) create a prior distribution using the Principle of Maximum Entropy, (2) update that distribution using any new information and Bayes theorem, and, (3) use this new distribution for inferencing [Jaynes, 88].

The fuzzy proponents argue that information is not crisp enough to be measured using the quantitative laws of probability. To overcome this difficulty, the concept of a membership function is used. It has yet to be determined for many researchers if there is any essential difference between using fuzzy information and exact numbers with probabilistic error bounds. At this point, many people agree that fuzzy information can be a useful concept for engineering systems and simplifying the code that runs those systems. It may turn out that it is a mathematical difference analogous to that between matrix and wave mechanics in physics.

> **CRITICAL POINT 6:** A full understanding of the relationship between various approximate ways of measuring information is needed.

Another important issue related to measuring uncertainty in information objects is the notion of entropy. That there is a relationship between information and entropy has been postulated for many years. A number of information measures have been proposed [Arndt, 01], including those by Shannon [Shannon and Weaver, 71] and Stonier [Stonier, 91]. Information is a measure of the decrease of uncertainty, and its representation requires an organized notation. Entropy is a measure of the increase of randomness. If one takes an organized body of information and randomizes it (adds noise) then there is less information and higher entropy.

The term “information” is itself used in different ways, however, because organization can mean many things. In thermodynamics, it is molecules behaving in an organized fashion. In Shannon’s communication examples, it is strings of symbols sent from a sender arranged in a way that can potentially lessen uncertainty at a receiver that can decode the symbols. In other uses, however, information has to be relevant to some task being performed by a system. In computation, it is related to complexity considerations. The work of Solomonoff, Kolmogoroff and Chaitin [Chaitin, 92] links information conveyed by symbols in logic and information systems with complexity of computation, and relates them to Shannon’s measures, as well.

The problem of information content is that it is “about *something*”. How do we compare information about two different subjects? The answer may be that we just do not do so, at least if the subjects are independent. But how do we know if they are independent? We may not want to mix oranges and apples; but, if we are concerned about fruit, we can develop information about them because they are no longer independent. Consider the following simple experiment. Suppose we have nine pieces of fruit, five oranges and four apples, and someone puts three of them in a bag. If we find three apples, we know that there are no oranges. This becomes much more difficult when we get to questionable or fuzzy sets - try repeating this experiment with five big apples and four small apples. This second experiment is typical of the problem of measuring information content. It depends on the individual system and its ontology. Independence can be classified as being in different dimensions, analogous to dimensions in physics; but, it seems there may be too many dimensions to measure.

> **CRITICAL POINT 7:** What we need to measure defines the model of the world that a system expects to find and its ways of coping with that world. The set of its behaviors may be infinite and unknowable if the machine is complex, but it is defined indirectly if we can predict behavior through the model. To predict behavior accurately, the information needs to characterize the ontology and what is done to information based on the ontology.

This point would seem to suggest an arduous task for satisfactory measurement of the relevant information that flows through a system; but it also suggest a potential benefit. Measurement of the information, if it is adequately powerful, can potentially "give back" to us in understanding enough value to repay the effort we put into creating and applying the metrics and techniques.

It is clear that every system that deals with information, whether computational or biological, contains an internal model of the outside world – an ontology. In computer programs, the elements of the ontology are data objects and procedures for manipulating those data objects; both are used by the program. Like matter and energy in physics, data objects and procedures in an ontology are related. Consider the notion of an ordered list of customers. It is a data object; yet it can be defined by a random set of customers and a sorting procedure. Its inputs are the (unordered) list and a statement of the type of order desired; and its output is the ordered list. If we want to integrate two systems that need an ordered list of customers for some purpose, it is important that we be confident that they employ the same order; otherwise they will not correctly operate together. To do that, it is necessary to measure the ontologies of the two systems to see if that is true.

The order example is not very complex on its surface. In practice, it is not possible, in general, to find out whether two algorithms producing an arbitrary order (not just a simple linear one) are outputting the same information. This is a consequence of a variety of undecidability results. So it is necessary to consider how we can use standards and measurements to be relatively confident that there is going to be interoperability between the two systems.

System Complexity and Information

The notion of providing appropriate level of information is also coupled to the type of decisions that are going to be made with that information. As such, the impact of the same information object will be different for different decision-makers. In the manufacturing context, the definition of performance measures for the system determines the type of decisions that must be made. Hence, a characterization of the amount of information needs to be developed in the context of the decisions being made and the impact of the information needs to be measured in terms of the performance measures. As a possible answer to this question, we explore the merger of two nascent theories, information theory and complexity theory.

CRITICAL POINT 8: Amount of information needed to make good decisions is directly proportional to the system complexity. Moreover, the impact of reduced information is higher in more complex systems.

Before we discuss the relationship between information, decisions, and complexity, we need to define the term *system complexity*. Despite the qualitative appreciation of the concept of *complexity* in manufacturing systems at the design, planning, and control stages, there is very limited fundamental understanding of the behavior of complexity, various factors influencing complexity, and the benefits of complexity in terms of improved system performance.

The term complex system has been discussed in system analysis literature. Extrapolating from different contexts in which the idea of complexity is used, a complex system may refer to one whose static structure or dynamic behavior is counterintuitive or unpredictable [Casti, 79]. It may refer to a system with patterns of connections among subsystems that make prediction of system behavior difficult without substantial analysis or computation. Or, it may refer to a system in which the decision-making structures make the effects of individual choices difficult to evaluate [Lofgren, 77; Simon, 81]. In many situations it is desirable to increase the number of components and interconnections in a system in order to increase the system's potential for achieving a higher level of performance. On the other hand, one might want to reduce the size of a system since a large system might be too difficult to understand, describe, or control, thereby resulting in a lower overall performance. Computational or algorithmic complexity is often used for classifying manufacturing planning and control problems [Garey, 79]. However, computational complexity does not capture all the aspects of complexity in manufacturing systems. In fact, the question, ``does a system fundamentally change or become simpler if a better algorithm is invented for solving the problem at hand?'', suggests the need for *system-related* complexity measures in addition to *algorithm-related* complexity measures. Also, computational complexity does not necessarily relate to the performance of the system, since computational complexity is an algorithm-related measure. For example, consider two heuristics used for solving the *n* job/*m* machine, job-shop scheduling problem, which is known to be *NP* hard. One cannot comment on the quality of solutions generated by these two heuristics just by comparing their computational complexities.

In recent years, several researchers have attempted to capture the essence of complexity in systems using information theoretic metrics. For example, [Deshmukh, 93] characterizes complexity of a system in terms of its static structure or time-dependent behavior. Static complexity can be viewed as a function of the structure of the system, connective patterns, variety of components, and the strengths of interactions. Dynamic complexity is concerned with unpredictability in the behavior of the system over a given time period. The manufacturing environment consists of physical systems in which a series of sequential decisions need to be made in order to produce finished parts. The sequence and nature of these decisions are not only dependent on the system capabilities but also on the products being manufactured in the system. Hence, any measure of system complexity should be dependent on both the system *and* the product information. The difficulty in making production decisions arises from the number of choices available at each decision point and the unpredictability of the effects of each choice on the system performance. Computational complexity can be considered the algorithmic effort required to evaluate these choices. In addition to these facets of complexity, there is another aspect that relates to the control of manufacturing systems. Static and dynamic complexities are discussed assuming a constant control scheme. However, different manufacturing systems mandate varying levels of control actions. [Deshmukh, 93] also suggests a control complexity measure for measuring the difficulty of controlling a manufacturing system. All the metrics proposed in this work essentially measure the *minimum amount* of information needed to describe the system or control actions. [Deshmukh, 93] also shows that the impact of lack of information is higher as the systems get more complex.

Hence, the definition of the appropriate amount of information and impact of that information on the system performance is of little use without explicit consideration of the decision being made using that information. This observation makes comparison of information objects even more difficult, but even more imperative.

CONCLUSION

We have argued that there is an intrinsic relationship between information and performance. In complex systems, where decisions are made by software applications, understanding this relationship is critical. We have argued further, that achieving this understanding depends on our ability to compare information objects. Yet, we do not know today how to measure equivalence of information nor its impact on a system. We need to be able to do so to evaluate existing systems, to engineer new systems, and, to integrate both. The benefits will be seen primarily in highly complex systems that utilize a large amount of knowledge from either other parts of the system or devices in the real world. On the other hand, should the promise of "ubiquitous computing" come true, information will permeate physical systems as well. In this paper, we have argued that the measurement of the ontology of a system is a fundamental part of realizing these goals.

REFERENCES

Arndt, C. Information Measures: Information and its Description in Science and Engineering, Springer-Verlag, Berlin, Germany, 2001.

Chaitin, G. J. Information-Theoretic Incompleteness, World Scientific, 1992 (Reprinted 1998.

Casti, J., Connectivity, Complexity and Catastrophe in Large-Scale Systems, John Wiley, New York, 1979.

Deshmukh, A., Complexity and Chaos in Manufacturing Systems, PhD Thesis, School of Industrial Engineering, Purdue University, West Lafayetts, IN, 1993.

Garey, M. and Johnson, D., Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, New York, 1979.

Jaynes, E., `The Relation of Bayesian and Maximum Entropy Methods (510Kb),' in Maximum-Entropy and Bayesian Methods in Science and Engineering, **1,** G. J. Erickson and C. R. Smith (eds.), Kluwer, Dordrecht, 1988

Pritsker, A., Papers, Experiences, Perspectives, Systems Publishing, New York, 1990.

Shannon, C. and Weaver W., The Mathematical Theory of Communication, University of Illinois Press, Urbana, IL, 1971.

Stonier, T., Information and the Internal Structure of the Universe, Springer-Verlag, Berlin, Germany, 1991.

Simon, H., The Sciences of the Artificial, The MIT Press, 1981.

Upton, D., The Operation of Large Computer Controlled Manufacturing Systems, PhD Thesis, School of Industrial Engineering, Purdue University, West Lafayetts, IN, 1988.