Research in Engineering Design (1999)11:116–131

Research in
Engineering
Design

A Theory of Complexity, Periodicity and the Design Axioms

Nam P. Suh

Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract. *One of the topics that has received the attention of mathematicians, scientists and engineers is the notion of complexity. The subject is still being debated, as it lacks a common definition of complexity, concrete theories that can predict complex phenomena, and the mathematical tools that can deal with problems involving complexity. In axiomatic design, complexity is defined only when specific functional requirements or the exact nature of the query are defined. Complexity is defined as a measure of uncertainty in achieving a set of specific functions or functional requirements. Complexity is related to information, which is defined in terms of the probability of success of achieving the Functional Requirements (FRs). There are two classes of complexity: time-dependent complexity and time-independent complexity. There are two orthogonal components of time-independent complexity, i.e.,* real complexity *and* imaginary complexity. *The vector sum is called* absolute complexity. *Real complexity of coupled design is larger than that of uncoupled or decoupled designs. Imaginary complexity can be reduced when the design matrix is known. As an example of time-independent imaginary complexity, the design of a printing machine based on xerography is discussed. There are two kinds of time-dependent real complexity: time-dependent* combinatorial complexity *and time-dependent* periodic complexity. *Using a robot-scheduling problem as an example, it is shown that a coupled design with a combinatorial complexity can be reduced to a decoupled design with periodic complexity. The introduction of periodicity simplifies the design by making it deterministic, which requires much less information. Whenever a combinatorial complexity is converted to a periodic complexity, complexity and uncertainty is reduced and design simplified.*

Keywords: Axiomatic design; Complexity; Design axioms; Periodicity

1. Introduction

On May 6 1997, the *New York Times*, one of the leading newspapers in the United States, carried an article entitled 'Researchers on Complexity Ponder What It's All About'. The appearance of such an article in a daily newspaper indicates that the issue of complexity has reached the center stage of science and technology in the 1990s. The article stated that

> Some of the grandest phenomena, like the coursing of comets around the sun, are marvelously predictable. But some of the most mundane, like weather, are so convoluted that they continue to elude the most diligent forecasters. They are what scientists call complex systems. Though made up of relatively simple units – like the molecules in the atmosphere – the pieces interact to yield behavior that is full of surprise[s].

1.1. Past Attempts to Define Complexity

In spite of all the efforts that have been made, mathematicians, scientists and engineers have not even accepted a common definition of what is meant by complexity. According to the author's colleague, Seth Lloyd[1] there are some three dozen different ways scientists use the word 'complexity'. Some definitions dealt with the complexity of process; for instance, how much computing it would take to solve a problem (Cover and Thomas 1991). Complexity has also been equated with the scale of measure – how many bits of information it takes to describe an object or a message (Shannon and Weaver 1949). Those in the field of manufacturing associate complexity with how much effort it would take to manufacture a product (Suh 1990). Chaitin (1987) and others came up with a concept called 'algorithmic complexity'. The basic idea is that simple tasks can be done by short computer programs and complex tasks, by longer programs. According to this view, one should be able to measure the complexity of a task by the

Correspondence and offprint requests to: N. P. Suh, Cross Professor and Department Head, Department of Mechanical Engineering, MIT 3-173, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA. Email: npsuh@mit.edu

[1]This statement was attributed to Professor Seth Lloyd of MIT by the *New York Times* article.

length of its most compact description. The problem with this idea is that the length of even the shortest computer program depends upon the design of the software as well as coding.

Gell-Mann proposed the idea of schema to identify the system's regularities as a means of defining complexity. He claimed that the length of the schema measures what he calls 'effective complexity', which is roughly the length of a compact description of the identified regularities of an entity. In the case of language, the schema is its grammar. Gell-Mann and Lloyd (1996) also proposed the concept of total information, which is effective complexity plus an entropy term that measures the information required to describe the random aspects of the entity. Bennett developed a different measure of complexity called 'logical depth'. The idea is to gauge how long it would plausibly take for a computer to go from a simple blueprint to the final product. Huberman and Hogg (1994) equates complexity with 'a phase transition' between order and randomness. Lloyd and Pagels (1988) equated complexity to free energy. There are many other views (Yates 1978, in Flood et al. 1993). All of these efforts are attempts to discover the basic absolute measure for complexity, which is contrary to the concept of information and complexity used in axiomatic design.

A very recent issue of Science[2] devoted a special section on the topic of Complex Systems. The journal dealt with complex systems in many fields of science, including life sciences, chemistry, mathematics, biology, physiology, geology, meteorology and economy. No attempt was made to present a unified definition of complexity. It is interesting to review the notion of complexity used by different authors to describe the complexity of their fields. Some of these notions are:

(a) In the introductory article by R. Gallagher and T. Appenzeller, 'complex systems' is taken to be one whose properties are not fully explained by an understanding of its component parts.
(b) In their article entitled 'Simple Lessons from Complexity', N. Goldenfield and L. P. Kadanoff states that ''complexity means that we have structure with variations. Thus, a living organism is complex because it has many different working parts, each formed by variation in the working out of the same genetic coding.''
(c) G. M. Whiteside and R. F. Ismagilov states the following in their article on *Complexity in Chemistry*: ''. . . a complex system is one whose evolution is very sensitive to initial conditions or to small perturbations, one in which the number of independent interacting components is large, or one in which there are multiple pathways by which the system can evolve. Analytical descriptions of such systems typically require nonlinear differential equations. Their second characterization is more informal, that is, the system is 'complicated' by some subjective judgement and is not amenable to exact description, analytical or otherwise.''
(d) In the abstract of their article entitled 'Complexity in Biological Signaling Systems', G. Weng, U. S. Bhalla and R. Iyengar states that ''Complexity arises from the large number of components, many with isoforms that have partially overlapping functions, from the connections among components; and from the spatial relationship between components.''

[2]These articles came out on the April 2 1999 issue after this paper was submitted for publication.

1.2. Axiomatic Design Perspective of Complexity and Information

Many of the past ideas of complexity are not consistent with that defined in axiomatic design. In many of the past works, complexity was treated in terms of an absolute measure. In axiomatic design, information and complexity are defined only relative to what we are trying to achieve and/or want to know. Information was defined as a logarithmic function of the probability of achieving the specified Functional Requirements (FRs), where the probability of achieving a specified FR was determined by computing the area under the system probability density function (pdf) within the common range (Suh 1990). Complexity is related to information.

To generalize the notion of complexity in the context of axiomatic design, we need to define the term complexity more precisely. We should be able to specify the meaning of complexity in the following situations:

1. What is the complexity in a bar of AISI 1020 mild steel that has to be machined to the dimensions of 1 meter in length, 0.02 meters in diameter, and 10 microns in surface finish?
2. What is the complexity of a machine that has seven (7) FRs?
3. What is the complexity of a laser printer?
4. What is the complexity of a manufacturing process designed to make Nylon fibers with a microcellular structure consisting of 1 micron diameter bubbles with a cell density of 10^{12} bubbles /cm3?
5. How complex is the job of being the weatherperson?

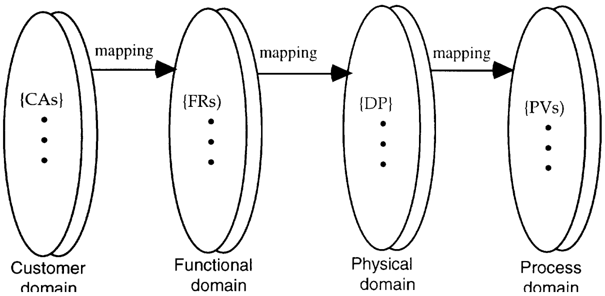

Fig. 1. Four domains of the design world. {x} are characteristic vectors of each domain. Design of products involves mapping from the functional domain to the physical domain. Design of processes involves mapping from the physical domain to the process domain.

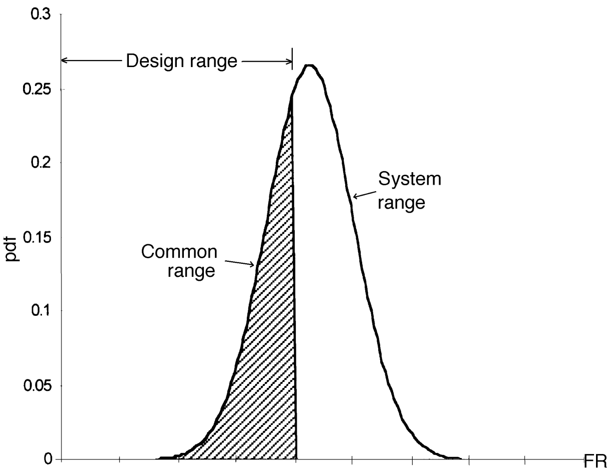

Fig. 2. Design range system range and common range, with probability of satisfying the FR given by the area under the system pdf in the common range (shaded area).

2. Complexity, Uncertainty, Information, and Periodicity

2.1. Preliminary Remarks

In axiomatic design, the design process is described in terms of the mapping between domains. The design goals for a product (or software, systems, etc.) are described in the functional domain in terms of Functional Requirements (FRs). The design task is to achieve the set of specified FRs by mapping FRs in the functional domain to Design Parameters (DPs) in the physical domain (see Fig. 1).[3] Thus, the selection of DPs determines the probability (and uncertainty) of satisfying the FRs.

[3]When the design matrix that relates the {FRs} vector to the {DPs} vector is diagonal, the design is defined as uncoupled. When it is triangular, it is a decoupled design. All others are coupled designs.

When the FR is defined, its desired target value FRo and its tolerance are specified in the design range as shown in Fig. 2. However, the actual pdf of the resulting design embodiment is the system range, which may be different from the design range[4].The portion of the design range overlapped by the system range is called the common range.

If the system pdf for a given FR is denoted p_s(FR), then the probability P of satisfying the FR is given by

$$P(dr^1 \leq FR \leq dr^u) = \int_{dr^1}^{dr^u} p_s(FR)d(FR) \qquad (1)$$

where the limits of integration, dr^l and dr^u, are the lower and upper limits of the design range, respectively.

Information content I is defined in terms of the probability P of satisfying a given FR as

$$I = -\log_2 P \qquad (2)$$

$$= -\log_2 \int_{dr^1}^{dr^u} p_s(FR)d(FR) \qquad (3)$$

The information content for satisfying a number of FRs for an entire system is just the sum of the information content I_i of the separate FR_i

$$I = \Sigma I_i \qquad (4)$$
$$= \Sigma - \log_2 P_i$$

where P_i is the probability of satisfying FR_i given by Eq. (1).

Because the system has a fixed number of FRs, complexity is unrelated to the number of FRs, but instead is the probability that a system will achieve all FRs. A system with low total I (i.e. high probability of achieving all FRs) is less complex than another system with exactly the same number of FRs and DPs, but with high total I (low probability of satisfying all FRs). This leads us to a specific definition of complexity.

2.2. Definition of Complexity

Complexity is defined as a measure of uncertainty in achieving the specified FRs.[5] Therefore, complexity is related to information content which is defined as a logarithmic function of the probability of achieving

[4]For detailed discussion of the design range and the system range, see Suh (1990, 1999).
[5]This definition is consistent with such ideas as a machine with many parts being more complex than machines with a fewer parts, since the uncertainty of achieving the machine functions increases with the number of parts. Uncertainty increases as the ability to predict the future outcome decreases.

the FRs. The greater the information required to achieve the FRs of a design, the greater is the information content (Suh 1999), and thus the complexity. These ideas will be expanded in this paper.[6]

There are two kinds of complexity: *time-independent* complexity and *time-dependent* complexity. Time-independent complexity can further be divided into time-independent *real* complexity and time-independent *imaginary* complexity. Time-dependent complexity may also be divided into two kinds: time-dependent *combinatorial* complexity and time-dependent *periodic* complexity. These complexities are examined and defined in this section.

2.3. Time-Independent Complexities: Real Complexity, Imaginary Complexity and Absolute Complexity

In axiomatic design, time-independent complexities – *real complexity* and *imaginary complexity* – are defined to deal with two kinds of uncertainties *real* uncertainty and *imaginary* uncertainty. Imaginary complexity is not at all related to real complexity; that is, it is orthogonal to real complexity. *Absolute complexity* is defined as a vector sum of these two orthogonal components of time-independent complexity.

Real Complexity is *defined* as a measure of uncertainty when the probability of achieving the FR is less than 1.0 because the common range is not identical to the system range. In Fig. 2, the uncertainty is given by the clear portion (un-shaded area) of the system range. Real uncertainty in design exists because the actual embodiment of the design does not quite satisfy the desired FR at all times.

The probability of achieving a given FR is determined by the overlap between the design range and the system range, called the common range (Fig. 3). Therefore, the real uncertainty exists even when the Independence Axiom is satisfied, if the common range is not the same as the system range. Thus, the real complexity can be related to the information content, which was defined in terms of the probability of success of achieving the desired set of functional requirements as

$$I = \sum_{i=1}^{n} \log_2 \left(\frac{1}{P_i} \right) \quad (5)$$

where P_i is the probability of achieving FR_i and n is the total number of FRs. Therefore, the information content given by Eq. (5) is a measure of uncertainty, and thus related to the real complexity. If we denote the real complexity as C_R, then we will define the real complexity to be equal to the information content as

$$C_R = I \quad (6)$$

When there are more than two FRs, coupled designs have larger real complexities than an uncoupled or decoupled design that satisfy the same set of FRs.[7]

Real complexity may be reduced when the design is either uncoupled or decoupled, i.e. when the design satisfies the Independence Axiom. For uncoupled designs, the system range for each FR can be shifted horizontally by changing the DPs until the information content is the minimum, since other FRs are not affected by such a change. Therefore, the mean value of FR provided by the system can be determined by adjusting the corresponding DP until the information is at a minimum. For decoupled designs, the system range can be shifted to seek the minimum information point by changing the DPs in the sequence given by the design matrix. The best values of DPs can be obtained by finding where the value of the real complexity reaches the minimum when the following two conditions are satisfied.

$$\sum_{i=1}^{n} \frac{\partial C_R}{\partial DP_i} = 0 \quad (7)$$

$$\sum_{i=1}^{n} \frac{\partial^2 C_R}{\partial DP_i^2} > 0 \quad (8)$$

When the design is uncoupled design, the solution to Eqs (7) and (8) can be obtained for each DP without any regard to other DPs, i.e. each term of the series must be equal to zero. In the case of decoupled designs, these equations must be evaluated in the sequence given by the design equation, since the design matrix is triangular.

[6]Gunnar Sohlenius in his unpublished note entitled 'Notes on Complexity, Difficulty and Axiomatic Design' argues that ''high information content should be used as measure of uncertainty, complication or difficulty rather than complexity.'' These differing definitions of complexity exist because in the daily usage of English 'complexity' is used to mean many different things. In this paper, the word 'complexity' has a specific definition.

[7]This is consistent with a theorem that states the following (Suh 1990, 1999): **Theorem 18** (Existence of an Uncoupled Design) *There always exists an uncoupled design that has less information than a coupled design.*

In the case of a coupled design, real complexity can also be changed, but the minimum information point for each FR is no longer meaningful, because when one of the DPs is changed to affect only one FR, all other FRs may change. Therefore, the minimum information point is defined only for the entire set of DPs where the sum of information for the entire set of FRs is the minimum. This corresponds to an ‘optimum’ point, which is often sought in operations research. However, this is a wrong design solution since many of the FRs can be satisfied exactly in the design space if the Independence Axiom is satisfied. In many cases of coupled design, Eq. (7) may never be satisfied.

Imaginary Complexity is defined as the uncertainty that is not real uncertainty, but arises because of the designer’s lack of knowledge and understanding of a specific design itself. Even when the design is a good design, consistent with both the Independence Axiom and the Information Axiom, imaginary (or unreal) uncertainty exists when we are ignorant of what we have.

To understand the distinction between real and imaginary uncertainty, consider a decoupled design with n FRs and n DPs given by the triangular matrix in Eq. (9).
which may be generally written as

$$\begin{Bmatrix} FR1 \\ FR2 \\ FR3 \\ \ldots \\ \ldots \\ \ldots \\ FRn \end{Bmatrix} = \begin{bmatrix} X0\ 0\ 0\ldots\ldots0 \\ X\,X0\,0\ldots\ldots0 \\ X\,X\,X0\ldots..0 \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ X\,X\,XX\ldots..X \end{bmatrix} \begin{Bmatrix} DP1 \\ DP2 \\ DP3 \\ \ldots. \\ \ldots. \\ \ldots. \\ DPn \end{Bmatrix} \tag{9}$$

$$\{FRs\} = [A^{LT}]\{DPs\} \tag{10}$$

where $[A^{LT}]$ is a lower triangular matrix.

The design represented by Eq. (9) satisfies the Independence Axiom, and thus the design can be implemented because there is no uncertainty associated with this design if the DPs are changed in the order indicated in Eq. (9). If the common range is the same as the system range, then the real complexity is equal to zero. If the common range is not the same as the system range, there is a real uncertainty and real complexity. This real complexity cannot be removed unless the system range and the common range are made to be the same by choosing new DPs, or by making the design more robust so as to remove uncertainty.

The decoupled design given by Eq. (9) can be a source of *imaginary uncertainty*, despite the fact that the design does satisfy the Independence Axiom. It is called imaginary uncertainty because the uncertainty is not real but the perceived uncertainty exists nevertheless. This imaginary uncertainty exists only in the mind of the designer, because the designer does not know that the design represented by Eq. (9) is a good design or when the designer does not write the design equation.[8]

Suppose the designer does not recognize that the design, although it can be represented by Eq. (9), is a decoupled design and does not know that the DPs must be changed in a proper order to make the design achieve the given set of n FRs. Therefore, the designer resorts to trial-and-error methods of evaluation, trying many different combinations of DPs to satisfy the FRs. Then, the probability of finding the right combination of n DPs to satisfy the entire set of FRs is given by[9]

$$P = \frac{1}{n!} \tag{11}$$

The probability of achieving FRs through a random trial-and-error process goes down rapidly with an increase in the number of FRs, as shown in Table 1. When n is 5, the probability of satisfying all five FRs is 0.008, which is a small number, and the information content is $\log_2 120 = 6.9$. Therefore, this design appears to be very complicated, and one would say that this design is very complex because the

Table 1.

n	$n!$	$P = 1/n!$
1	1	1
2	2	0.5
3	6	0.1667
4	24	0.04167
5	120	0.8333×10^{-2}
6	720	0.1389×10^{-2}
7	5,040	0.1984×10^{-3}
8	40,320	0.2480×10^{-4}

[8]Imaginary complexity exists under many other circumstances – whenever the perceived complexity is not entirely due to real complexity, there exists imaginary complexity.

[9]The actual probability of satisfying the FR may be smaller than the probability of finding the right combinations since the system range for each FR may be different from the design range. However, for large n, this probability of finding the right combinations is likely to dominate.

uncertainty is large. However, it is not the case of real uncertainty; this uncertainty is artificially created as a result of the lack of understanding of the system designed or how nature really works. Therefore, this kind of uncertainty is defined as *imaginary uncertainty*. In many situations, this imaginary uncertainty leads to the erroneous conclusion that a design is complex although it may not be – all due to the lack of fundamental understanding the axiomatic design theory.

If we denote the imaginary complexity as C_I, then it may be related to the probability of success given by Eq. (11) as

$$C_I = \log\left(\frac{1}{P}\right) = \log n! \qquad (12)$$

For very large n, e.g. $n > 100$, Eq. (12) may be written as

$$C_I \approx n(\log n - 1) \qquad (13)$$

If the design matrix is such that there are m possible combinations of n DPs that can equally satisfy the FRs, then the probability of satisfying the FRs is given by $P = m/n!$. Therefore, as m increases, the design will appear to be less complex because the imaginary uncertainty decreases. However, the real uncertainty does not change with m.

Example 1. Xerography-based Printing Machine. HG Company is one of the leading printing press manufacturers in the world. They just developed a commercial label-printing machine based on xerography technique. This machine can quickly print commercial labels as soon as the original copy is inserted into the machine since it is based on xerography principle. The design of the machine is schematically illustrated in Fig. 3.

An optical image of the label is transmitted to the surface of the selenium-coated aluminum cylinder using light. The cylinder rotates at a constant speed. When the charged section of the cylinder passes by the toner box, the oppositely charged liquid-toner transfers to the charged part of the selenium surface. To control the thickness of the toner layer on the selenium drum, the wiper roll removes the extra thick toner layer from the surface of the cylinder. Paper is fed into the gap between the main selenium cylinder and the paper feed roll. When the paper comes in contact with the selenium surface under the light pressure exerted on the paper by the paper feed roll, the image is firmly printed on the paper.

The Advanced Engineering Division of HG Co., which was developing this printing machine, ran into trouble. They found that sometimes the selenium coating is badly scratched, creating poor images and damaging the expensive selenium coated rolls (about $4000 per cylinder, which was about 18 inches in diameter). Since the beta machine had to be shipped in a few months, they assigned many scientists and engineers to figure out the problem and solve it.

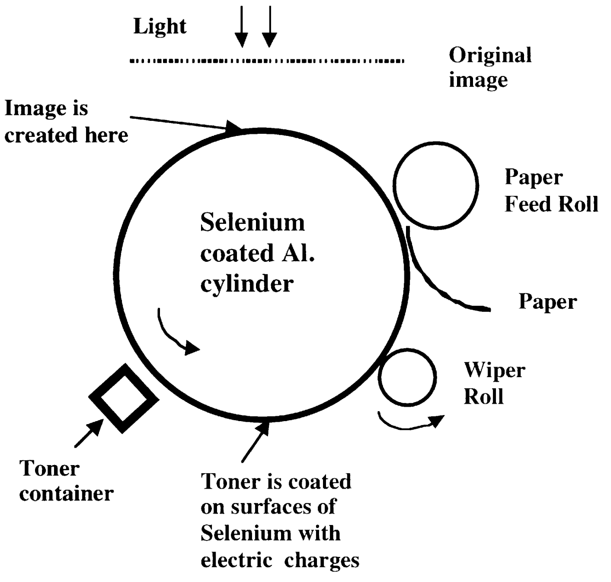

Fig. 3. Schematic drawing of the xerography-based printing machine. Image is transmitted to the selenium coated aluminum cylinder using light. When the charged section of the cylinder passes by the toner box, the oppositely charged toner liquid transfers to the charged part of the selenium surface. The wiper roll removes the extra thick toner layer from the surface of the cylinder. Paper comes in contact with the selenium surface under the light pressure exerted on the paper by the paper feed roll.

The scientists and engineers came to the conclusion that the scratch marks (in the form of lines) must have been a result of abrasive wear. They attributed the source of abrasion to be unknown abrasive particles that somehow got into the toner tank. Their reasoning received much internal support from everyone in the Advanced Engineering Division, since the machine (which was about 30 feet long) was being assembled at a corner of a large machine shop. They conjectured that tiny metal chips from the machining operation somehow got into the tank, occasionally scratching the selenium drum.

To make sure that the toner was free of any abrasive particles, they installed special filters that would remove all particles greater than a few microns and put plastic sheet around the machine to create a clean environment around the machine. However, the despicable scratch marks would not go away! The high-level managers of the company became uneasy about the situation, and decided to consult a tribologist at MIT about this abrasive wear problem.

The tribologist told them to read a reference book on tribology to learn all about the things that affect abrasive wear.

After a few months, the tribologist received an urgent call from HG Co. They said that they have to ship the beta machine to a customer's factory in a week and yet the scratch marks were still there – apparently the reference book did not do any good! The tribologist was asked to hop on an airplane right way and visit the factory where the machine was being tested. So he went.

What do you think the tribologist found at the HG Company?

Solution. The tribologist, who also knows something about axiomatic design, listened to HG engineers and scientists who explained all the things they had done and their theory on the cause of the problem. They were sure that somehow devilish small particles are getting into the printing machine and the toner box and these particles caused the scratches on the surface. Indeed, the examination of the surface and micrographs indicated that the scratch marks were typical scratches caused by abrasive particles. However, the tribologist was not convinced that the explanation given by HG engineers and scientists were correct.

The functional requirements of the machine, assuming that abrasive particles somehow got into the toner box, may be chosen to be the following:

FR1 = Create electrically charged images
FR2 = Coat the charged surface with toner
FR3 = Wipe off the excess toner
FR4 = Make sure that abrasive particles do not cause abrasion
FR5 = Feed the paper
FR6 = Transfer the toner to the paper
FR7 = Control throughput rate

The tribologist reasoned that DPs used by HG personnel (although they did not use axiomatic design) in their trial-and-error approach are as follows:

DP1 = Optical system with light on selenium surface
DP2 = Electrostatic charges of the selenium surface and the toner
DP3 = Wiper roller
DP4 = Filter
DP5 = Paper feeding mechanism
DP6 = Mechanical pressure
DP7 = Speed of the cylinder

Since there are seven FRs and seven DPs, there are more than 5000 combinations if they try to run the tests by trying different combinations of DPs. The probability of success of achieving the FRs by a trial-and-error method is quite small. Even if they devised an orthogonal array experiment, there are still too many tests to determine the cause. Furthermore, if the design is a decoupled design, simply identifying important DPs through the orthogonal array experiment will not yield the answer. Indeed, their extensive tests did not yield any solution!

The design matrix that may represent the thinking of the HG engineers may be represented as

	DP1	DP2	DP3	DP4	DP5	DP6	DP7
FR1	X	0	0	0	0	0	0
FR2	X	X	0	0	0	0	0
FR3	0	0	X	0	0	0	0
FR4	0	0	X	X	X	0	0
FR5	0	0	0	0	X	0	0
FR6	0	0	0	0	0	X	0
FR7	0	0	0	0	X	0	X

According to the above design matrix, the order of FR4 and FR5 as well as DP4 and DP5 should be changed to obtain a triangular matrix. What the matrix is saying is that if the paper feeding mechanism or process creates particle, filtering the toner outside the machine will not do any good. The filter must also remove the particles generated by the paper feeding mechanism. This is not easy to achieve.

Another solution is to prevent large particles from ever approaching the interface by means of controlling the fluid motion. For abrasion to occur, kinematic consideration indicates that somehow the abrasive particle, whatever it may be made of, must be stationary at the interface between the main cylinder and the wiper roll. If the particle goes through, then at most, the selenium surface will be indented rather than scratched. Then FR4 may be decomposed as

FR41 = Prevent the abrasive particle being anchored at the interface between the main cylinder and the wiper roll
FR42 = Prevent the particles from approaching the interface

At this point, it is instructive to consider the kinematics and fluid mechanics of the toner motion

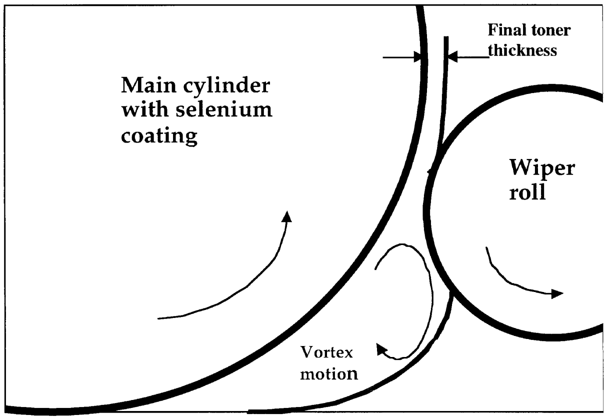

Fig. 4. The vortex motion of the toner and the rotational direction of the main cylinder and the wiper roll.

near the entrance between the wiper roll and the main roll.[10] When the machine is first started, if the main cylinder rotates first before the wiper roller is rotated, the toner will be dragged along and any particle in the toner will anchor at the narrow section of the opening between the roller and the main cylinder. Furthermore, if the surface speed of the main cylinder is greater than that of the counter-rotating wiper roller, the pressure at the narrow gap will be greater, and the tendency to squeeze in the abrasive particle at the interface between the main cylinder and the wiper roller will be greater. On the other hand, if the wiper roller starts turning first and if the surface speed of the wiper roller is greater than and opposite (as indicated in the figure) to the surface speed of the main cylinder, then the pressure at the entrance will be less. It will reduce the tendency for large particles to come into the narrow gap. Furthermore, the vortex motion in the toner will prevent the large particles from approaching the main cylinder/wiper interface as shown in Fig. 4.

Then, DPs may be chosen as

DP41 = The order of rotation of the wiper roller and the main cylinder (wiper roller rotates first)

DP42 = The surface speed of the wiper roller greater than and opposite to the surface speed of the main cylinder

The tribologist made the suggestion that DP41 and DP42 be implemented. The machine had a digital control system, and therefore DP41 and DP42 could be implemented immediately. He also asked the HG engineers to put abrasive particles into the toner intentionally. When the machine was turned on, there were no more scratch marks! The tribologist happily hopped on an airplane and returned to Boston. He had spent six working hours at HG Company to solve the problem, while many months using the trial-and-error approach prior to his visit produced no success.

If a design is uncoupled with a diagonal design matrix and zero information content, both the real uncertainty and the imaginary uncertainty are equal to zero. In this case, both those who do and those who do not understand axiomatic design may come to the same conclusion on complexity and uncertainty.

*Absolute Complexity.*The absolute complexity C_A is defined as

$$C_A = C_R + jC_I \tag{14}$$

C_R, the real complexity, and C_I, the imaginary complexity, may be plotted in a two-dimensional complex plane, as shown in Fig. 5. The '*j*' is the imaginary unit and the '*j*' axis is for the imaginary complexity. C_I and C_r are orthogonal to each other, because the imaginary complexity has no relationship to the real complexity, and vice versa. The vertical '*j*' axis is the axis of 'ignorance', since it is caused by the lack of knowledge, which yields the perception that the design is more complex than it really is. The horizontal axis represents real uncertainty as a result of design and/or unknown behavior of nature. The absolute complexity C_A is shown as the vector sum of C_r and C_I, since C_r and C_I are orthogonal to each other.

It is difficult to predict the exact values of C_r and C_I *a priori* if the design is coupled or decoupled. However, a bound for C_I can be estimated if the

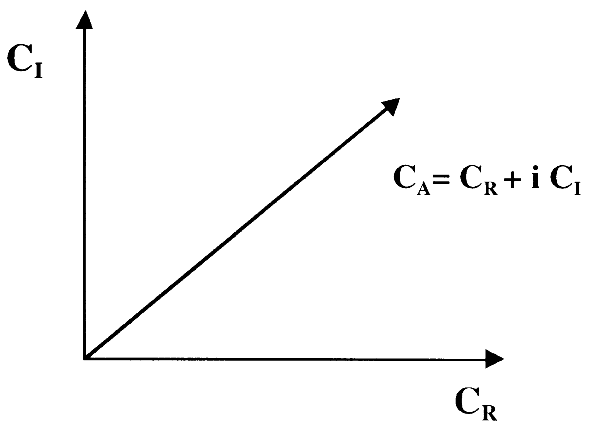

Fig. 5. Complexity consists of the real and imaginary part. The vertical axis is the axis of ignorance and the horizontal axis represents real uncertainty as a result of design and/or unknown behavior of nature.

[10]In selecting DPs, the designer's knowledge of associated physics and engineering is obviously indispensable. Axiomatic design cannot make up for the lack of fundamental understanding of physics, mathematics, and other associated knowledge base. Either designer must know the fundamentals or a database must be provided.

design is a decoupled design using Eq. (12). When the design is uncoupled, the imaginary component of complexity is equal to zero, and only real complexity may exist if the system range is not inside the design range. In the case of coupled design, the magnitude of the imaginary complexity can be very large and dominate the real complexity.

Based on the foregoing discussion of absolute complexity, real complexity and imaginary complexity, we may adopt the following definition of complexity:

> Time-independent complexity is a measure of uncertainty in achieving a given set of FRs and thus related to information content. It consists of two orthogonal parts – the real and the imaginary complexities. The real complexity is defined as a measure of real uncertainty in achieving a given set of FRs and thus related to the information content given by Eq. (5). The imaginary complexity – perceived uncertainty – is caused by the designer's lack of knowledge about the system designed or the behavior of nature.

2.4. Time-Dependent Complexity: Time-Dependent Combinatorial Complexity and Time-Dependent Periodic Complexity

The time-independent complexity discussed in the preceding section dealt with complexity involved in making design decisions because of the uncertainties inherent in the system designed, which is defined as the real complexity, or because of uncertainties caused by the lack of knowledge, i.e. ignorance, which is defined as the imaginary complexity.

There is another kind of uncertainty – time-dependent uncertainty – because the future events occur in unpredictable ways, and thus cannot be predicted. In this section, the time-dependent complexity will be defined. In the next section, the means of reducing the time-dependent complexity through the use of the Independence Axiom and the Information Axiom will be discussed.

Time-dependent complexity arises because in many situations, the future events cannot be predicted *a priori*. Many of these problems are combinatorial problems that can grow complicated indefinitely as a function of time, because the future events depend upon the decisions made in the past but in an unpredictable way. In some cases, this unpredictability is due to the violation of the Independence Axiom. An example is the problem associated with scheduling a job shop. Job shops are typically engaged in machining a variety of parts that are brought to them by their customers. In this case, the future scheduling – which parts are produced using which machines – is affected by the decisions made earlier and the complexity is a function of the decisions made over its past history. This type of time-dependent complexity will be defined as time-dependent *combinatorial complexity*.

There is another kind of time-dependent complexity: *periodic complexity*. Consider the problem of scheduling airline flights. Although airlines develop their flight schedules, there exist uncertainties in actual flight departures and arrivals because of the unexpected events such as bad weather or mechanical problems. The delayed departure or arrival of one airplane will affect many of the subsequent connecting flights and arrival times. However, since the airline schedule is periodic each day, all the uncertainties introduced during the course of a day terminate at the end of a 24-hour cycle, and hence this combinatorial complexity does not extend to the following day. That is, each day the schedule starts all over again, i.e. it is periodic, and thus uncertainties created during the prior period are irrelevant. However, during a given period there are uncertainties due to combinatorial and other uncertainties. This type of time-dependent complexity will be defined as time-dependent *periodic complexity*.

Both time-dependent combinatorial and time-dependent periodic complexities are real complexities.

In the next section, it will be shown that a time-dependent combinatorial complexity can be changed to time-dependent periodic complexity, greatly reducing information content, uncertainty, and ultimately, complexity. This is done through re-design or by introducing decouplers.

3. Reduction of Uncertainty – Conversion of a Design with Time-Dependent Combinatorial Complexity to a Design with Time-Dependent Periodic Complexity

The Independence Axiom and the Information Axiom can be used to reduce the information content of a design, deal with time-dependent combinatorial complexity, and to convert a combinatorial complexity problem to a deterministic problem through the introduction of periodicity.

Example 2 is a beautiful case that shows how a coupled design was decoupled by applying the Independence Axiom so that the robot schedule would not affect the manufacturing process. This decoupling is achieved by adding decouplers. This example shows how a time-dependent combinatorial complexity problem was reduced to a periodic complexity problem, reducing the information needed to make the system work to a minimum and increasing the reliability of the system by removing uncertainty.

Example 2. From Combinatorial Complexity to a Periodic Complexity – Design of Fixed Manufacturing Systems for Identical Parts (adopted from Oh and Lee 1999)

Highest-level design of a fixed manufacturing system. Consider the case of making identical parts by processing them through a set of different processes. For example, it may be related to coating, curing, and developing a photoresist material – a viscous substance that is light-sensitive which is used to take images on silicon-wafer surfaces – for semiconductor manufacturing. In this case, the highest-level FR may be stated as follows:

FR1 = Maximize the return on investment (ROI)

To maximize ROI, we have to produce the maximum number of coated wafers, sell them at the highest possible price, minimize the manufacturing cost, and minimize the capital investment. However, we will consider here only the task of maximizing the output of a dedicated, automated machine. Then, the design parameter may be written as

DP1 = Dedicated automated machine that can produce the desired part at the specified production rate.

The FRs of the dedicated and automated machine may be written as

FR11 = Process wafers in various modules

FR12 = Transport the wafers between modules, between the loading dock and modules, between modules and unloading dock

The corresponding DPs may be chosen to be the following:

DP11 = Process modules

DP12 = Robots

The Cs are

C1 = Throughput rate

C2 = Manufacturing cost

C3 = Quality of the product

C4 = Yield (production of acceptable products divided by the total output)

When a machine with a robot and process modules was designed, FR11 and FR12 were coupled. For example, sometimes the wafers from two or more different modules would be finished nearly at the same time, and demand transport to the next module by the robot. However, since one robot cannot perform two functions at the same time, a decision had to be made as to which wafer is to be picked up first. The decision affected all subsequent decisions. The original design solved this problem by using the ‘if, then’ type expert system algorithm. Sometimes incorrect decisions were made delaying the operation, and the machine would come to a complete stop when there was not an appropriate ‘if, then’ rule. The problem with this design is that it is a coupled design. The design equation for this coupled design is given by

$$\begin{Bmatrix} FR11 \\ FR12 \end{Bmatrix} = \begin{bmatrix} X & X \\ X & X \end{bmatrix} \begin{Bmatrix} DP11 \\ DP12 \end{Bmatrix} \tag{15}$$

Therefore, a decision was made to design the robot schedule based on axiomatic design so that the Independence Axiom is satisfied. The new proposed design can be expressed using the design equation

$$\begin{Bmatrix} FR11 \\ FR12 \end{Bmatrix} = \begin{bmatrix} X & 0 \\ X & X \end{bmatrix} \begin{Bmatrix} DP11 \\ DP12 \end{Bmatrix} \tag{16}$$

Equation (16) expresses the fact that in the proposed design, DP11 (the process modules) will affect both FR11 (process wafers) and FR12 (transport wafers), but DP12 (the robot) will not affect FR11 (process wafers). This is an important design decision – the design will be done so that the robot motion shall not affect the processes. All subsequent decisions as we decompose these FRs and DPs must be consistent with this decision. Design represented by Eq. (16) states that, given an arrangement of the modules, we must design the transportation system that will not affect the manufacturing process. This is a decoupled design.

Since this machine processes exactly identical parts, a ‘push’ system may be designed, where the part will be supplied to the machine on a regular time interval T. T is equal to (3600/m) seconds, where m is the number of wafers supplied to the machine per hour. T is the cycle time during which the robot must pick up wafers from all modules at least once. The number of modules needed for each process is related to the period T, since if the process time of a module is larger than T, it will take more than one module to be able to meet the required throughput rate. If the process time in Module i is denoted as $t_{P,i}$ (sec), the number of the modules needed to meet the production requirement, n_i, is given by

$$n_i = Int \frac{t_{P,i}}{T} = Int\left[\frac{tp, i}{(3,600/m)}\right] \tag{17}$$

Int [x] is a function that rounds x up to the next nearest integer. The total number of modules, M, required to process the wafers is given by

$$M = \sum_{i=1}^{N} n_i \tag{18}$$

where N is the number of tasks, i.e. processes. These process modules must be arranged so that the robots can serve all of these modules in the shortest possible time.

Within the cycle-time period T, the module for each process (or one of the modules when there is more than one module for a given process) completes its task. Therefore, within a given period T, the robot must pick up the wafers from these modules that just completed their process cycles and transfer them to the next set of modules. The robot must also deliver a wafer from the supply cassette station to the first module, as well as from the last module to the outgoing cassette station, all in a given period T. If it takes t_T for the robot to transport the water from one module to the next, then the number of moves the robot can make in time T is equal to T/t_T.

The sequence of the robot operation is as follows. The robot picks up a wafer from the supply cassette station and delivers it to Module 1 for Process 1. Upon completion of Process 1 in Module 1, the robot picks up the wafer from Module 1 and inserts it into Module 2 for Process 2. Similarly, from Module 2 to Module 3, and so on. When it is again the time to pick up another new wafer from the supply cassette after an elapse of time T, the robot goes back to the supply cassette and loads from the cassette to the first module for Process 1. If the first Module 1 is still processing a wafer, this new wafer is loaded into the second Module 1. This sequence of wafer transfer continues until the entire process is completed. In one period T, the robot must move all the wafers that have just finished a prescribed process and move it to the next module for another process. [Note: there can be more than one module for each process, as per Eq. (30).] In addition, the robot must load a new wafer from the supply cassette station and also deliver the finished wafer from the last module to the outgoing cassette station.

A conflict can arise in scheduling the robot motion if two processes are completed nearly at the same time (i.e. within the time required for single robot motion), since the robot has to be at two different places at the same time. This coupling of functional requirements can cause a system failure. In the past, this problem was tackled using the 'if, then' type of algorithm for deciding which wafer the robot should pick up next. An 'if , then' type of AI approach is unreliable, since the number of combinations increases continuously, as each subsequent decision depends upon the decisions made earlier. The number of possible combinations increases to Πn_i where n_i is the number of modules available for each process. Furthermore, when there is no appropriate 'if, then' rule, the system breaks down.

This problem can be solved rigorously by introducing decouplers, i.e. by redesigning the system! The coupling occurs when two or more wafers complete the prescribed processes nearly simultaneously (within the transport time of the robot). We can decouple the pick-up functions by introducing 'decouplers' – devices that store the wafers until the robot becomes available.[11] The role of decouplers is to decouple functional requirements of the transport. The decouplers do not have to be separate physical devices. In this case, the modules can act as decouplers by letting the wafers stay in the modules longer. Decouplers provide queues between modules so that the wafers can be picked up in a predetermined sequence by the robot. The design task is to determine where the decouplers should be placed, and how long their queue should be. Some modules cannot act as decouplers if the process time in the module is tightly controlled for chemical reasons.

When decouplers are introduced with queue q_i, the process time T_C increases. As T_C increases, the number of modules may increase depending on the process time t_P of each module. Therefore, we have dual goals: decouple the process by means of the decouplers and minimize T_C by selecting the best set of q_i. Then, FR12 (transport wafers) may be decomposed as

FR121 = Decouple the process times
FR122 = Minimize the number of modules M

The corresponding DPs are

DP121 = Decouplers with queues, q's
DP122 = The minimum value of T_C

The design equation is given by

[11] The term 'decoupler' was used for the first time by J. T. Black (1991).

$$\begin{Bmatrix} FR121 \\ FR122 \end{Bmatrix} = \begin{bmatrix} X & 0 \\ X & X \end{bmatrix} \begin{Bmatrix} DP121 \\ DP122 \end{Bmatrix} \tag{19}$$

To minimize the number of modules M, we must satisfy the following two conditions:

$$\sum_{i=1}^{N} \frac{\partial M}{\partial q_i} = 0$$
$$\sum_{i=1}^{N} \frac{\partial^2 M}{\partial q_i^2} > 0 \tag{20}$$

where N is the number of processes.

Analytical solution for queues in decouplers.[12] Having designed the manufacturing system, we must replace those X's with mathematical expressions if they can be modeled. In this section, the queues q's will be determined through modeling and analysis to determine the exact relationship between FR121 (decouple the process times) and DP121 (decouplers with q's).

If we denote the time the wafer has to be picked up upon the completion of process j in Module i as T_i, then T_i is the sum of the process time t_P and the accumulated transport time t_T, which may be expressed as

$$T_i = t_P + t_r \tag{21}$$

T_i, t_P and t_T are normalized with respect to the sending period T, i.e. actual time divided by T. Therefore, throughout this analysis, all of the times will be dimensionless, i.e. the actual time divided by the period T.

Since the total process t_P is the sum of the individual process times, $t_{P,j}$, and the transport times, t_T, is the sum of the all robot transport time, $t_{T,j}$, Eq. (21) may be expressed as

$$T_i = t_P + t_T = \sum_{j-1}^{i} t_{P,j} + \sum_{j=1}^{i-1} t_{T,J} \tag{22}$$

The number of pick-up moves the robot can make in a given period T is given by

$$n_R = \frac{T}{t_T} \tag{23}$$

If there are N process steps, there are M modules with wafers that have gone through their respective processes and ready to be picked up within a given period T. Within this time period, the robot must pick up all these wafers from the modules that completed the process. The robot must pick up a wafer at time τ_i that is measured from the beginning of each period T, which may be expressed as

$$\tau_i = T_i - Int\left(\sum_{j=1}^{i} t_{P,j} = \sum_{j=1}^{i-1} t_{T,j}\right)$$
$$= \sum_{j=1}^{i} t_{P,j} + \sum_{j=1}^{i-1} t_{T,j} - Int\left(\sum_{j=1}^{i} t_{P,j} + \sum_{j=1}^{i-1} t_{T,j}\right) \tag{24}$$

where $Int(x)$ is a function that rounds x down to the next nearest integer. However, if the pick-up times are coupled because two or more processes are finished nearly at the same time, the robot cannot implement the schedule given by Eq. (24).

We must modify the design to decouple the process by adding decouplers with queue q_i. For example, Process 1 in Module 1 and Process 3 in Module 3 are finished within the transport time required $t_{T,1}$, then the robot cannot pick-up both pieces at the same time. Therefore, in this case, we may add a 'decoupler' to Module 3, which may be either a physically separate device or just queue in Module 1 to keep the wafer in longer. In this case, T_i given by Eq. (22) is extended by q_i. Then, the new time for pick-up T_i* is given by

$$T_1^* - T_1 = q_1$$

Similarly, extending it to a more general case,

$$T_2^* - T_2 = q_1 + q_2 \tag{25}$$
$$T_3^* - T_3 = q_1 + q_2 + q_3$$
etc.

Substituting these relationships into Eq. (24), we obtain the modified actual pick-up time. If we denote this modified time as τ^*_i, then (τi $-\tau^*_i$) may be expressed approximately[13] as

$$\tau_i^* - \tau_i = \sum_{j=1}^{i} q_j = a_{ij} q_j$$

where the matrix a_{ij} is defined as

$$a_{ij} = \begin{Bmatrix} 0 & when\ i < j \\ 1 & when\ i > j \end{Bmatrix} \tag{26}$$

[12] This robot scheduling problem comes from SVG, Inc. which hired many consultants to solve the problem without obtaining any satisfactory solution. While Dr Larry Oh, Vice President of SVG, Inc. and the author were waiting at an airport, the author suggested the use of decouplers and Dr Oh (with my graduate student Tae-Sik Lee) came up with this elegant closed form solution. A patent has been filed by SVG to protect this work.

[13] The approximation neglects the possibility that the integer function can change its value when argument of the function is close to an integer. However, the exact solution has the same result as shown by Eq. (26) because it is off-set by an integer.

We can approximately determine τ_i* by solving Eq. (25), by determining where the decouplers may be needed, and by approximating the values of queues.

Equation (26) may be expressed as

$$\{\tau_i - \tau_i^*\} = [a_{ij}]\{q_j\} \tag{27}$$

where $\{x\}$ denotes a vector and $[x]$ is a matrix. Equation (27) may be solved for q_i as

$$\{q\} = [a]^{-1}\{\Delta\tau\} = \frac{1}{|a_{ij}|}[A]\{\Delta\tau\} = [A]\{\Delta\tau\} \tag{28}$$

where

$\Delta\tau = \tau_i - \tau_i^*$

$[a] \equiv$ matrix with elements a_{ij}

$|a_{ji}| \equiv$ determinant of matrix$[a] = \prod_{i=1}^{N} a_{kk} = 1$

$[A] = Adj[a_{ij}] = [A_{ji}]$

$A_{ji} = (-1)^{i+j}M_{ji}$

$M_{ji} \equiv$ minor of a_{ji}

Equation (28) can be solved iteratively. To solve Eq. (28), we need to know $\{\Delta\tau\}$, which can be approximated by estimating reasonable values for τ_i* and by solving Eq. (26) for τ_i. The value for τ_j* can be estimated by adding transport time to τ_i* since $|\tau_i^* - \tau_j^*| > t_T$, for all j's except $j = i$. The solution can be improved by successive substitution of the improved values of τ_i*. The determinant $|a_{ij}|$ of the triangular matrix $\{a\}$ is equal to the product of the diagonal elements.

Since the best solution is the one that makes the total cycle time T_C a minimum, we must seek for a set of values of q_i that yield a minimum value for the total queue, Σq_i. When the precise control of processing time is critical, the queue q_i associated with the process should be set to equal to zero.

To solve Eq. (28) for the best set of queues q's, Oh (1998) and Oh and Lee (1999) developed an optimization software program based on genetic algorithms. Multiplying these q's by T, we can obtain actual values of queues.

Determination of the queues of a fixed manufacturing system that processes identical parts. A manufacturing system is being developed for coating of wafers. To produce the final semiconductor product, wafers coated with photoresist must be subjected to various heating and cooling cycles at various temperatures for different duration before they can be shipped to the next operation. The manufacturing system is an integrated machine that consists of five process steps involving five different modules. A robot must place the wafer with the coatings into these modules, then take them out of the modules, and transport them to the next process module according to a preset sequence. We want to maximize the throughput rate by using the robot and use the modules most effectively. The desired throughput rate is 60 units an hour. A constraint is the use of a minimum number of modules. The time it takes for the robot to travel between the modules is 6 seconds. The wafers are processed through the sequence shown in Table 2.

The process times in Modules B and E must be precise because of the critical nature of the process. The cycle time is assumed to be the process time plus the transport time both for placement and pick-up of the wafer.

The robot must pick up the wafers from a supply bin (load-lock) and deliver it Module A and when the process is finished, it must pick up the wafer from Module E and place it on a cassette. These operations take 6 seconds each.

Solution. The minimum number of modules is dependent on the process time T_C and the desired throughput rate. The required number of modules is as follows:

Modules	*Number of modules*
A	2
B	1
C	2
D	2
E	1

Without any decouplers, there are simultaneous demands for the service of the robot as shown in Fig. 6, which shows the time the process is finished in each of the modules within a given period T. In the figure, the horizontal axis is dimensional less time – one (1) represents one period T. Since the transport time is equal to 6 seconds, i.e. (T/10), the figure shows that Processer 1 and 3 are completed so close to each other that the robot has a conflict. Similarly, Processes 2 , 3 and 5 are all finished nearly at the same time.

Table 2.

Steps	Modules	Temp. (C)	Duration ± tolerance (Seconds)
1	A	35	50 + 25
2	B	80	45 +/- 0
3	C	10	60 + 20
4	D	50	70 +10
5	E	68	35 +/- 0

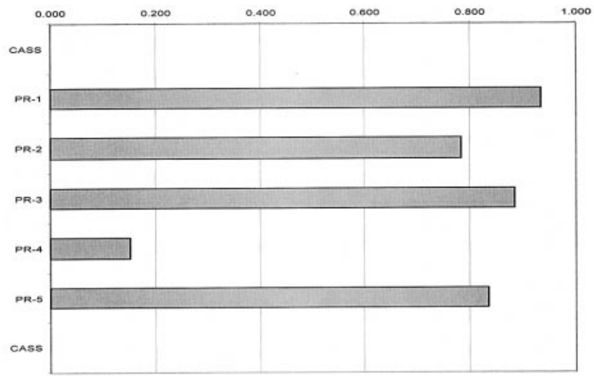

Fig. 6. The pick up schedule in a period T without any decouplers. There are conflicts among the processes finished in Modules 1,2, 3 and 5. This result is obtained using the software program developed by Oh (1998) and Oh and Lee (1999).

The solution is obtained by solving Eq. (26) using the software program developed by Oh and Lee (1999). The best solution was obtained by finding a set of values that give the shortest cycle time T_C solving Eq. (26) repeatedly, and using a genetic algorithm. The solution yields the following values for q's:

q_A = 0.2 sec
q_B = 0 sec
q_C = 11.0 sec
q_D = 10.2 sec
q_E = 0 sec

The queues for B and E are zeros, since the tolerance on these two modules is specified to be zero. Therefore, the queues of other modules has been adjusted to make these two queues to be zero. The actual pick-up times at the completion of the processes of Modules A, B, C, D and E are given in Fig. 7.

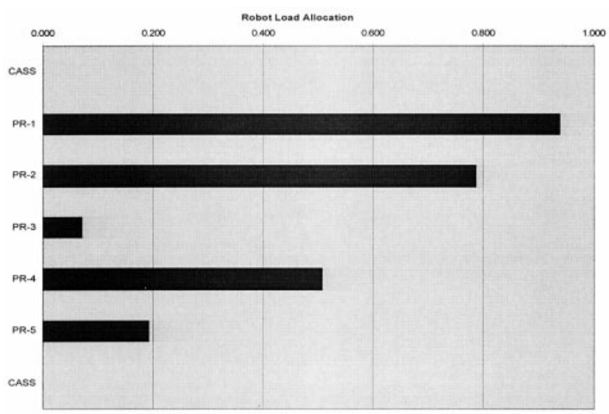

Fig. 7. The actual pick-up times of Modules A, B, C, D and E. PR-1 is Process 1 that takes place in Module A, PR-2 is for Module B, PR-3 is for Module C, PR-4 is for Module D, and PAR-5 is for Module E. This solution is obtained using the software program developed by Oh (1998) and Oh and Lee (1999)

There are other possible sets of solutions for q_i, but they may not give the minimum M or minimum T_c and the minimum Σq_i.

One of the interesting results of this solution is that the number of combinations for part flow reduces down from several thousands to a few, because the parts flow through the manufacturing system along deterministic paths. What the concept of decouplers has done is to change a combinatorial problem into a periodic function that repeats itself with a given cycle that is deterministic.

In the robot-scheduling problem discussed so far, the scheduling problem was changed from a combinatorial problem to a deterministic one, immensely reducing the uncertainty and complexity. Furthermore, the infinite time-dependent combinatorial problem was changed to a periodic problem where the cycle within the sending period was made to repeat itself by adding decouplers. This is an important consequence of applying the Independence Axiom to these random events to create a 'periodicity'. This change of the task with uncertain outcome to one with a definite outcome reduces uncertainty and makes the task much less complex. In other words, the introduction of decouplers has introduced periodicity and changed a combinatorial problem into a deterministic problem.

It is a very significant finding that this creation of 'periodicity' reduces, if not eliminates, uncertainty and thus, complexity associated with an infinite combinatorial complexity. Furthermore, an infinite time-dependent combinatorial system cannot be sustained because the uncertainty associated with its future events becomes too large. The system then becomes risky and unreliable. This means is that even when it is not clear as to how a period can be defined, it is better to stop an event and restart with new initial values to reduce the uncertainty of future events, where the current decisions affect future events and probabilities. Nature forces this periodicity by giving a finite life to all living beings. These observations which are extensions of the Independence Axiom and the Information Axiom can be stated as a theorem[14]:

[14]Other theorems can be found in Suh (1990, 1999).

Theorem 26 (Conversion of a system with infinite time-dependent combinatorial complexity to a system with periodic complexity). Uncertainty associated with a design (or a system) can be reduced significantly by changing the design from one of the serial, time-dependent combinatorial complexity to a periodic complexity one.

This relationship between complexity and periodicity has many important applications and implications.

4. Distinction between Time-Independent and Time-Dependent Complexity

One of the interesting questions is whether there is any generalization that can be made of the relationship between the time-independent and the time-dependent complexity. Although no thorough investigation has been made of this issue, it seems that they are distinct from each other. It is as distinct as the elliptic partial differential equation is different from the hyperbolic partial differential equations.

In the case of time-independent complexity, the end result is governed by the given set of FR and DP relationships. This is in contrast to the case of time-dependent complexity, which depends upon the initial condition, but unless the system goes back to the same set of initial condition periodically, the distant future behavior is totally unpredictable. That is, in the case of time-dependent complexity, the initial condition has little control over the long-term behavior of the system – unlike the case of hyperbolic or parabolic partial – differential equations. In the time-dependent complexity case, the initial condition is not distinguishable from the state at the system at any other time in terms of its ability to control the long-term behavior.

5. Other Implications of Periodic Complexity – A Speculation[15]

5.1. Nature and Living Beings

One of the important discoveries this paper described is the power of changing a design with the time-dependent combinatorial complexity with a design that has periodic complexity. It reduces uncertainty. When uncertainty is large, the future outcome cannot be assured. The periodicity also renews the life-cycle, and increases reliability of the system by re-starting the system from the same initial conditions over and over again.

It is interesting to note that nature has known this fact all along. Many things in nature are periodic. Atomic structure is periodic. The animal life-cycle is periodic. Most animals sleep daily to renew themselves. The life of all living beings is periodic and finite.

Nature sustains life by renewing itself periodically. If living beings would live forever, they will go through mutations and other changes that cannot be predicted *a priori* nor controlled. Therefore, all living beings stop living when these unanticipated events occur. They sustain and renew themselves by reproduction through the combination of the fundamental building blocks from ground zero periodically.

If we extend this speculation one step further, nature may deal with the continuing level of environmental pollution by replacing the current expansion of combinatorial complexity of nature with a periodic one, on a grand scale. This will happen if the earth cannot support the current form of living beings without starting all over again. One may conclude that to prevent this unpleasant event from happening, human beings must discover a means of renewing the nature through less pollution.

5.2. Artificial Systems

The implication of Theorem 26 also has important implications on political and societal systems. A kingdom or a country that is ruled by a dictator without any possibility of renewal is one of the systems with time-dependent combinatorial complexity rather than a design with periodic complexity. Therefore, such a political system can undergo unexpected mutations, since the future outcome cannot be controlled or predicted *a priori*. Hence, the system can corrupt and deteriorate in a totally unexpected manner. There are certainly many historical examples of such systems, one of the most recent ones being the Soviet Union.

We must introduce periodicity to even political systems so that they can reduce uncertainty and renew themselves. Possible renewal mechanisms are periodic elections, a periodic setting of budgets and periodic auditing.

Universities must also be designed to have a periodic complexity. The existing mechanisms are academic semesters, fixed periods of study, and the academic tenure systems for faculty. Contrary to popular view, the tenure system at leading universities

[15]The topics discussed in this section may be classified as intellectual speculations, since they are not supported by any proof and/or experimental confirmation. Some of the speculative ideas are given as food for thought.

guarantees the renewal of academic life. It provides a means of renewing faculty on a regular cycle (typically 6–8 years), since many who are judged to be less than the best end up leaving the institution, although wrong decisions are sometimes made like many other human decisions. For the tenured faculty, the system depends upon the retirement system for periodic renewal. The fact that some universities have no longer the mandatory retirement system is not good in terms of periodic complexity.

6. Concluding Remarks

This paper has examined the issue of complexity, information, and uncertainty based on the Independence Axiom and the Information Axiom. It was shown that there are many different kinds of complexities.

In the time-independent situations, it was shown that there are two kinds of complexities: real complexity and imaginary complexity, which are orthogonal to each other. Absolute complexity is defined to be a vector sum of the real and the imaginary complexities.

In the time-dependent complexity arena, it was shown that there are two different kinds of complexities: combinatorial complexity and periodic complexity. In a system that is subject to combinatorial complexity, the uncertainty of the future outcome continues to grow as a function of time and as a result cannot have a long term stability and reliability. In the case of systems with periodic complexity, the system is deterministic and can renew itself over each period. Therefore, a stable and reliable system must be periodic. Starting from the application of the Independence Axiom, it was shown how a coupled system was be decoupled through design changes and how a combinatorial complexity problem could be changed into a periodic complexity design problem. A theorem was presented.

A case study was presented to show how the complexity could be reduced by redesign and by replacement of a combinatorial complexity problem with a periodic complexity one.

Finally, the consistency between nature and Theorem 26 (Conversion of a system with infinite time-dependent combinatorial complexity to a system with periodic complexity) is discussed. It was shown that many things in nature are periodic, consistent with the need to change a combinatorial complexity design to one of periodic complexity system to reduce uncertainty. It was also argued that the theorem may apply to political and societal systems as well. Periodic renewal of political systems and societal systems is essential for long-term self-sustainability of the system.

Acknowledgements

It is the pleasure of the author to acknowledge helpful discussions and reviews of this paper by Dr Larry Oh, Tae-Sik Lee and Jason Hintersteiner. The author is also indebted to Carol Vale of Ford Motor Co. for her valuable comments and editorial insights. The MIT research in axiomatic design has been supported by Silicon Valley Group, Inc., and the National Science Foundation, and the Charles Draper Laboratory, Inc. The support of Papken Der Torossian, William Hightower, Byung-Ho Ahn, and Dr. Ming Lou are deeply appreciated.

References

Black JT (1991) The Design of the Factory with a Future. McGraw Hill, New York

Chaitin GJ (1987) Algorithmic Information Theory. Cambridge University Press, Cambridge

Cover T M (1991) Elements of Information Theory. Wiley, New York

Flood RL, Carson ER (1993) Dealing with Complexity An introduction to the Theory and Application of Systems Science. Second edition. Plenum Press. New York

Gell-Mann M, Lloyd S (1996) Information measures, effective complexity, and total information, Complexity, 21:44–52

Huberman BA and Hogg T (1994) In: Nørretranders, T (ed) Märk världen. En bok om vetenskap och intuition. Bonnier Alba. Stockholm

Lloyd S, Pagels H (1988) Complexity as thermodynamic depth. Ann Phys 188:186–213

Oh HL (1998) Synchronizing Wafer Flow to Achieve Quality in a Single-Wafer Cluster Tool. Unpublished Report, Silicon Valley Group, Inc. (Patent Pending 1998)

Oh HL, Lee T-S (1999) Synchronous algorithm. International Conference on Axiomatic Design MIT June 2000 (accepted for presentation)

Shannon CE, Weaver W (1949) The Mathematical Theory of Communication. University of Illinois Press, Urbana, IL

Suh NP (1990) The Principles of Design. Oxford University Press, New York

Suh NP (1999) Axiomatic Design: Advances and Applications. Oxford University Press (to be published)