Design Theory and Practice I: A Critical Review of General Design Theory

Y. Reich

EDRC 12-45-91

Design Theory and Practice I: A critical review of General Design Theory

Yoram Reich
Engineering Design Research Center
Carnegie Mellon University
Pittsburgh, PA 15213

Keywords: design theory, General Design Theory, scientific method, research methods

Abstract: The method of conducting research significantly influences its progress and quality. This study advocates for subscribing to the scientific method of performing research, namely, generating theories, conducting experiments to support or refute them, and refining the theories. The arguments in favor of this method are discussed in the two parts of the paper. Part I critically reviews General Design Theory (GDT), a formal theory of design, and discusses its relation to building experimental design systems. Part II of the study describes an experimental system that is built on the foundation of GDT and its experimental testing. The experimental system and its performance are used to suggest modifications to the theory and extensions to the experimental system itself. These important recommendations conclude one successful scientific cycle, thereby supporting the benefits from adopting the scientific method.

1 Introduction

The method of conducting research significantly influences the research progress and quality. Two major methods for conducting research are: theoretical and experimental. The goal of this study is to show that the combination of the two, which is called the scientific method, is the most fruitful to pursue. First, we describe the three methods mentioned; then, we briefly discuss the scope of design addressed in this study and the organization of the study.

The theoretical method. The theoretical method is based on mathematical principles. The entities manipulated in this method are axioms or assumptions, theorems, and proofs[1]. The benefits of using a theoretical method are manifold. First, it requires the precise specification of the assumptions underlying the theory. Second, the objectives of the theory must be described precisely in the form of theorems. Third, the proofs of the theorems are based on concise derivations and produce unquestionable results. The only objection against the results is their potential scope, which heavily depends upon the scope of the axioms or assumptions.

The outlined benefits seem to outweigh any possible deficiency that may characterize this method. Even if the scope of the state-of-the-art theory is restricted compared to the scope of the real world, which is in fact the state of the design theory discussed later in this paper, there may still be important principles that can be abstracted from the theory and further studied experimentally. This tie between theory and experimental work is crucial as discussed later.

Despite the importance of a theoretical, in-depth understanding of design activities, design theories (that are based on mathematics) are very scarce. A notable exception is General Design Theory (Yoshikawa, 1981; Tomiyama and Yoshikawa, 1986; Tomiyama et al., 1989) which is critically reviewed in this paper.

One of General Design Theory (GDT) goals is to lead to good design of intelligent CAD systems. This concern is in correspondence with the need to test theories in realty. Unfortunately, GDT has not been seriously experimentally tested or compared with an experimental system. In addition, it has not been used or adopted by other researchers or practitioners than its developers.

In contrast to GDT, that is highly articulated as a mathematical theory, some of the other theories of design are only elaborate classifications of design problems (Eder, 1987; Rohatyński, 1983) or prescriptions of how design should be performed (Hubka and Eder, 1990; Warfield, 1990). These theories are hard to test. In sum, design theories are rare and had almost no influence on the experimental method described next.

The experimental method. In experimental studies, difficult problems in design research orpractice are tackled, solutions are generated and tested in experiments. Usually, no principled guidance is exercised in the generation of solutions; therefore, the activity is unstructured. In reality, experimentalists adopt a certain solution paradigm (e.g., use of grammar) and use domain knowledge to guide the solution generation; thus introducing limited structure into the research activity.

[1]This is still in line with Dixon's statement that theories are arrived at inductively (Dixon, 1987). If the theory is to have any impact on the real world, the axioms and the assumptions that are the basis of the theory must be inductive statements about the world. The consequences from these axioms and assumptions can and are deduced by mathematics.

We use the term solutions and not hypotheses, and solution paradigms and not theories, because there is a difference **between** the two sets of terms. Solutions are generated for addressing a single problem. Researchers often attempt to articulate statements about the broader applicability of the solutions; rarely are these statements tested. In contrast, hypotheses are statements meant to be general and tested in several experiments.

Most engineering design research follows the experimental method, and in many cases only in a deficient manner. The deficiency lies in the way solution are tested in design research. It is rarely that results are shown on a large variety of problems and almost never are performance statistics provided. Usually, a single demonstration is accepted as a sufficient evidence of a successful solution.

The scientific method. In the scientific method, hypotheses are created and tested in experiments. The process of hypotheses generation is crucial to the success of this approach. Design theories can provide a good mechanism for generating hypotheses, so are psychological studies of human designing. Since there is no guarantee that hypotheses are correct[2], replication of results, refinement of hypotheses and their refutation are necessary ingredients of this method.

The scientific method addresses two concerns raised against the two previous methods. The first concern criticizes the theoretical method for being restricted in scope and consequently irrelevant to applied research and practice. The second concern points to the lack of coherent guidance in experimental research. The combination of the two methods can resolve these criticisms. Similar benefits from a method that combines design theories and experimental design systems are discussed by Hongo (1985).

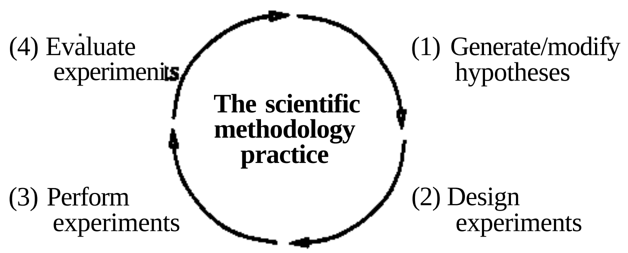


Figure 1: The scientific method cycle

Figure 1 illustrates the scientific method cycle. In step (1), hypotheses are generated from inductive observations about the world. In steps (2) and (3), experiments that attempt to refute or support the

[2]**In fact, hypotheses can never be proved. They are only generated to be refuted (Popper, 1965).**

hypotheses **arc designed and** executed. These steps constrain **the** generation of hypotheses in step (1), only **testable hypotheses arc** considered appropriate. Step (4) evaluates the results, leading to the refinement of **the original** hypotheses in a new application of step (1).

Design research is mostly experimental. However, research methods are not mature and often the principles of testing, replication, and refutation are not exercised. In many cases, a descriptive account of the work perfonned is provided, but without reporting much systematic testing. If test replications are not performed, it will not be sufficient that a hypothesis is the product of a theoretical statement (e.g., design is a hierarchical refinement) for the research to be sound. The contribution of such work to the collective knowledge of design cannot be assessed. This situation calls for a major change in research methods for design.

The scientific approach is lately overlooked in engineering design research (Antonsson, 1987). A quote appearing in (Dixon, 1987) summarizes the state of (experimental) design research methodology:

> "Proposals rarely advance theories or hypotheses. When they do, it is rarely a testable theory or hypothesis. When it is testable, actual testing is rarely proposed. When a test is proposed, it is rarely well conceived."

Emerging scientific method in design. There are some fragments of work in design that signify the emerging of scientific method in design research. They consists of theoretical statements induced from experience and studies that show the applicability of these theoretical statements to design.

Suh's two axioms (Suh, 1984) are one example that can be viewed as hypotheses about the quality of designs. The two axioms have evolved from a larger set of 7 hypothetical axioms (Suh et al., 1978) through research on the topic (Suh, 1984)—a process demonstrating the strength of the scientific method. The axioms **arc stated to** be used in synthesis, but they arc best used for comparing between designs by measuring how **well do** the designs satisfy the axioms. The two axioms arc used to derive 9 corollaries that provide elaborate guidelines for the generation of better designs.

Taguchi's method for quality design (Taguchi, 1980; Taguchi, 1986) is another example of scientific work that can also be perceived as a hypothesis about the quality of designs. This method is stated to be used in redesign. For a given precise function and developed structure of a design, an analysis is performed to result in parameter selections that minimize some objective function (e.g., minimize the cost to the customer due to malfunctioning). A study comparing both Taguchi's and Suh's approaches (Filippone, 1989) exemplifies work required in the scientific method forevaluating competing hypotheses or theories.

Scope of design. After articulating the terms in the opening paragraph it is important to introduce the interpretation of the terni "design" used in this study. Design is defined as a process. Given a description of a desired function and constraints, called *specification*, provide a description of an artifact that produces the function and satisfies the constraints. This description is called *artifact description* or *artifact structure*. We sometimes use the term *function* to refer to the specification.

Design is viewed as a sequential process consisting of five tasks (Reich, 1990b): *problem analysis* which includes the elaboration of the problem statement and the formulation of a search space; *synthesis* which is the process of search in the space of alternative configurations; *analysis* which is a verification that the synthesized candidate satisfies the specification; *redesign* which modifies the design until it satisfies the specification; and *evaluation* which augments the design by additional subjective judgment. The same structure of tasks appears in different design phases such as preliminary or detail design.

Organization. This study is divided into two parts. Part I, which appears in this paper, critically reviews a specific theory of design: General Design Theory (GDT) (Yoshikawa, 1981). It illustrates the definitions and theorems of GDT through an example domain. The paper articulates the consequences of GDT for building CAD systems and outlines how GDT can be implemented. The contribution of this part is two-fold. First, it allows understanding the theory without the need to be an expert in the theory of topology. Since the paper, however, does not compromise conciseness, the mathematical concepts are explained and illustrated with examples. This allows appreciating the second contribution of the paper, namely, the criticism of the restricted scope of the theory. In addition, this part outlines how GDT can be partially implemented and tested. This is further detailed in the second part of the study.

Part II of the study (Reich, 1991b) reviews an experimental design system, BRIDGER (Reich, 1991a), that was developed following the guidelines set by GDT (Reich, 1990a). This part compares the theory (GDT) with the experimental system (BRIDGER) and shows the significant similarities in their overall approach to design. This aspect corresponds to step (2) in Figure 1. BRIDGER was evaluated in several domains and shown increasingly better performance as it continued to learn from experience. The tests with BRIDGER corresponds to step (3) in Figure 1. The evaluation of the results and the analysis of the discrepancies between GDT and BRIDGER (correspond to step (4)) point to research directions that may broaden the scope of GDT and improve the performance of BRIDGER (correspond to step (1) in the second cycle).

Together, both parts of the study demonstrate that the scientific method of conducting design research offer several benefits. First, it focus work on crucial issues, thereby accelerating the development *oi* increasingly better design support systems. Second, the scientific approach provides a means for growing a collective body of knowledge about design that can be tested and refined. Note that we do not attempt to justify GDT or BRIDGER as approaches to design. Rather, we concentrate on the synergy between these approaches as manifested in the scientific method and the benefits that emerge from using this method in addition to those provided by each of the approaches separately. These benefits establish the support for subscribing to the scientific method.

The remainder of the paper is organized as follows. Section 2 describes the domain of chairs which is used to illustrate the ideas discussed in this paper. Section 3 reviews the concepts of GDT, emphasizing the important assumptions and results of the theory. Section 4 discusses the theory and its implications for building CAD systems and Section 5 summarizes the paper.

2 The domain of chairs

Figure 2 depicts eight chairs which arc used to explain the concepts and ideas discussed in this study. Each chairs in the figurc is denoted with a letter. The chairs provide *some functionality* that is summarized in Table 1. Each row describes a different function of a chair. The + in the table denotes that a chair provides the corresponding function, and a - denotes its lack thereof. Additional functions that chairs may have but that are not mentioned include *comfort, access to ceiling, resistance to fire,* etc. In addition to providing functions, each chair has properties that can be observed and therefore describe the *attributes* or the *structure* of the artifact; some of these are summarized in Table 2. Additional observable properties that chairs may have include *color, material texture, type of upholstery, dimensions,* etc.

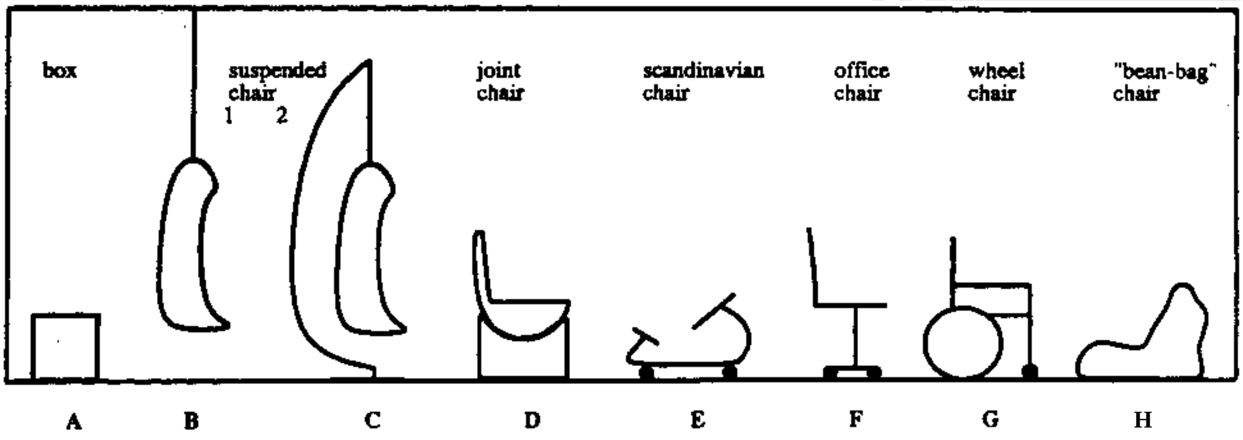


Figure 2: Chairs

Naturally, there are functions that are directly derived from the structure of a chair. For example, a chair

Table 1: Functional properties of chairs

	function	A	B	C	D	E	F	G	H
		chair							
1	**seating**	+	+	+	+	+	+	+	+
2	**back support**	+	+	+	+	+	+	+	+
3	**revolving**	—	+	+	+	+	+	+	+
4	**movable**	+	—	—	+	+	+	+	+
5	**stably support back**	—	+	+	—	—	+	+	—
6	**ordinary** design	+	—	—	—	—	+	—	—
7	contemporary design	—	+	+	+	—	—	—	—

Table 2: Observable properties of chairs

	structure	A	B	C	D	E	F	G	H
		chair							
1	**has seat"**	+	+	+	+	—	+	+	—
2	**has back support**	—	+	+	+	—	+	+	—
3	**has legs**	—	—	+	—	—	+	+	—
4	**has wheels**	—	—	—	—	+	+	+	—
5	**has vertical rotational dof**	—	+	+	+	—	+	—	—
6	**has light weight**	+	+	—	+	+	+	+	+
7	**is hanging**	—	+	+	—	—	—	—	—
8	**has stopper**	—	—	—	—	—	—	+	—

* **A seat** is **a horizontal stiff object.**

that ***has wheels*** **is** ***movable*** **or a chair that has a** ***vertical rotational degree*** *of* ***freedom*** *(dof)* is *revolving.* **Note that this structure-function relation may be an approximation; for example, chair C with** a rotational **dof does not allow for** 360° rotation. Other functions may be more complex and could no be inferred **from one observable property.** To illustrate, a chair can provide *back support* although it does not have a back. For instance, chair A provides back support due to its location near a wall. Its function is context dependent. Also, chair E provides back support due to its complex structure although is does not have a *physical* back support. Some functions may qualify other functions. For example, the function *stable back support* qualifies the function *back support.* This function is quite complex to assess. Chairs B and C are stable due to static considerations, whereas chairs D and H are not stable; chairs A, F, and G are structurally stable; and chair E does not even have a physical support.

All the previous examples have concentrated on inferring potential functionality from artifact structure. This is useful in *analysis.* We are mainly interested in *synthesis* which is the generation of artifact structure that will satisfy a desired function. For example, the specification of a chair that will be *movable* and *stably support back* leads to two potential designs F and G. These designs can be generated in two ways. The first way starts with *{A, D,* £, F, G,*H}* as the *movable* designs and refines them with the *stably support back* property. The second way starts with *{*B_y*C,F, G}* as the *stably support back* designs and refines them with the *movable* property. The most concise description of the solution is chairs that have *physical back support* and *wheels.* Of course, this description can lead to chairs not depicted in Figure 2. The design process was made easier by the use of the eight representative chairs as mediating between the specification and the design description. In the absence of these chairs, the process is much more difficult.

As an another example, assume that in addition to the previous specification, it is also required that the chair will have a *contemporary design.* There is no chair that satisfies these three functions. A *redesign* process can be invoked by taking the current candidate designs {F, G} and retracting either the *movable* or the *stably support back* specification properties and then trying the new specification. Since we are working with extensional description of the candidate designs, there will still not be any candidate that satisfies all the three specification properties. Nevertheless, we will have three sets of *nearly* good candidates: (1) *stably support back* and *contemporary* {B,C}, (2) *movable* and *contemporary {D}*$_{\%}$ and (3) *movable* and *stably support back {F,G}.* If the set of designs was not confined to the eight chairs, the second group could be made *stably support back* by adding a stopper. In contrast, the first group cannot be easily made movable: *B* is attached to the ceiling and *C* would have to receive wheels and a stopper.

The chair domain describes issues in the preliminary design stage. It does not discuss the selection of material and production techniques nor the sizing and proportions of various chair parts, both have substantial influence on the aesthetic and stylistic value of chairs. These issues are briefly raised in the theory. The incorporation of many more properties (both functional and observable) can elaborate the domain as seen in (Eastman, 1991), which discusses comfort analysis of chairs using a detailed model that better approximates real chairs.

As we show later, the chairs domain, in its present form, does not satisfy the axioms of GDT. Nevertheless, examples are extracted to show the meaning of GDT's definitions and theorems. In some cases, examples that violate the axioms or theorems are presented.

3 General Design Theory

General Design Theory (GDT) is one of the most formal theories of design. It attempts to cast design in the framework of set theory. GDT starts with assumptions about the nature of objects which are inductive generalization about the world; in fact, they are overgeneralizations. Then , GDT attempts to prove theorems about the nature of design. GDT makes interesting claims about design. As such, it tries to be a descriptive theory of the world, but more importantly, a prescription for the future development of computational support systems.

Unfortunately, GDT is not fully appreciated although at least 10 years have past since its presentation in English (Yoshikawa, 1981). Possibly, the perceived mathematical complexity prevented researchers from looking at the relevance of GDT to real design.

This section reviews GDT's terminology, definitions, assumptions, and theorems while using the chairs domain as a working example throughout. In many cases the exact wording of the definitions/axioms/theorems is maintained. While reviewing GDT, the section points to the most important assumptions, how they determine the success of the theory (measured as the ability to prove theorems about design), and how they can be relaxed.

3.1 Preview

The purpose of this preview is to motivate some of the concepts discussed by GDT. Since the theory of topology can be viewed as a generalization of the concept of continuity (Sutherland, 1975); it is natural to give a motivation from the area of continuous real functions.

For the purpose of simplicity, assume that in a certain design domain, a specification is described by a real number and the artifact is described by a real number as well. A design process is a mapping from the specification to the artifact description, hence can be described as a mapping/ :/?—•/?, where *R* is the set of real numbers.

Figure 3 illustrates an example domain. A function from the artifact description to the specification (i.e., analysis) is modeled as 1 + sin ʃ from 0 to *a,* and by an arbitrary function that is asymptotic to -oc at *d.* The mapping from the specification to the artifact description (i.e., synthesis) is the "inverse" of this function.

There are several important properties of continuous functions that are of interest in various design tasks such as synthesis, analysis, or redesign: *distance, continuity, convergence,* and *transformation.*

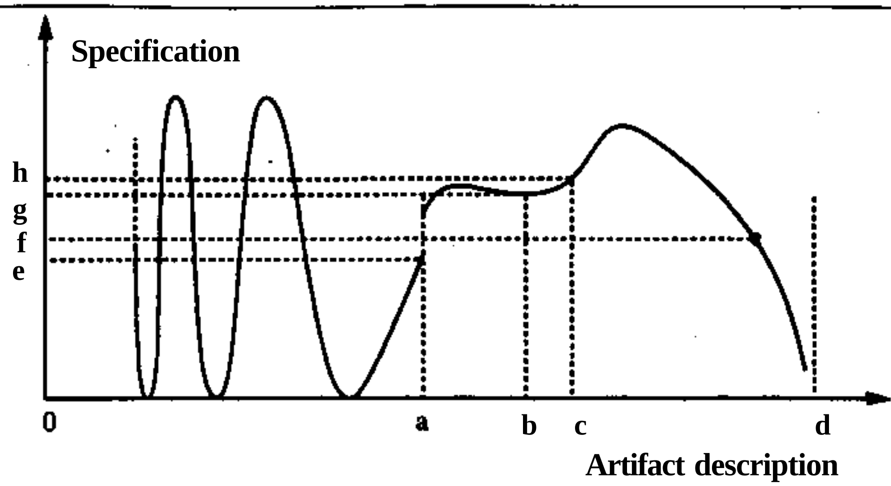


Figure 3: A simplified design domain

A *Distance* between two specifications or between two artifact descriptions is a useful concept. For example, distance can be used in evaluation. It can provide answers to questions such as: how close arc two specifications/functionalities, or how far is the functionality of the candidate from the desired functionality? In our example, both specifications and artifact descriptions arc described by real numbers, therefore, distances can be calculated by subtraction.

Continuity is a process oriented concept; it intuitively relates to the concept of distance. Continuity guarantees that a small change in the design description will result in a small change in the artifact functionality and vice versa. Therefore, if the current candidate differs slightly from the required function, a small modification to the structure may be sufficient.

For example, a required move from function *g* to *h* can be achieved by a small move from *b* to c. An artifact near 0 cannot be analyzed since the function is discontinuous at 0; therefore an artifact cannot be synthesized. A required move from *e* to/ (around point a, which is a discontinuity point) cannot be achieved by a small change in the artifact.

Convergence is also a process oriented concept; it provides a different perspective to continuity. Convergence guarantees that a sequence of incremental changes (e.g., from *b* to *c)* will cause only small incremental refinement changes (e.g., leading from *e* to/). A negative example are the small changes when moving from *a* to 0, or from *c* to *d* that do not converge to a specific artifact description.

A *transformation* that conserves the continuity or convergence properties is useful in design domains because it allows obtaining a different viewpoint of the specification and the partial design description

that may simplify future design steps. In topology, such a transformation is called *homeomorphism.* These useful concepts appears in a generalized form in topology. As such, topology provides an interesting perspective of viewing design. With this motivation we turn to reviewing General Design Theory.

3.2 Preliminary definitions

DEFINITION 1: An *entity* is a real object that existed, exists presently, or that will exist in the future.

EXAMPLE : Any chair that existed since the invention of the first chair and that will exist is an entity. For the purpose of simplification assume that Figure 2 contains all these entities.

DEFINITION 2: The set of all objects is called the *entity set*

EXAMPLE : The chairs domain can be viewed as an entity set.

DEFINITION 3: An *attribute* is a physical, chemical, mechanical, or any other property that can be observed by scientific means. Each entity has *values* for its attributes.

The terms property and value are not defined by GDT but can be given mathematical meaning (as suggested by T. Yagiu in the discussion following (Yoshikawa, 1981)). Note that usually property and attribute are synonymous. However, in GDT, the term attribute has a special meaning; it is a property that can be observed.

EXAMPLE : The properties listed in Table 2 arc all attributes of chairs. They all can be observed or measured by some scientific means. The table specifies the attribute-value pairs for each chair.

DEFINITION 4: When an entity is subjected to a situation, it displays a behavior which is called a *functional property.* The collection of functions observed in different situations is *the functional description* of the entity.

EXAMPLE : The properties listed in Table 1 are all functional properties of chairs. The table specifies the functional behavior manifested by each chair. For example, *(un)stable back support* is a behavior manifested when a human leans at the back, e.g., chair D will swing backward, therefore, it is not stable. It is clear that this function is a direct consequence of the joint (i.e., circular) seat-base connection; in most cases the function-structure relationships is much less clear. These relationships are very important in design.

A real object cannot be reasoned about in the mind of designers. Designers operate on an internal representation of the real objects.

DEFINITION 5: The representation of an object is called *concept of entity*.

We will refer to it as *entity* as well since from now on, only representations arc discussed, not real objects.

EXAMPLE : The representation of a chair using the function and structure properties from Tables 1 and 2 is a concept of entity. Also, a representation of a chair using its picture in Figure 2 as an icon is a concept of entity.

The discussion has concentrated on single entities and their representation. The important concept of *classification* is now introduced.

DEFINITION 6: A classification over the entity set is a division of the entities into several classes. Each class is called an *abstract concept.* The set of *all* abstract concepts is denoted by T.

EXAMPLE : Chairs A and B can form a class and the remaining chairs can form another class. A more *meaningful* classification can be obtained by classifying the chairs based on their properties. For example, the property *has legs* divides the set of chairs into two classes: chairs with legs (C, F, and G) and chairs without legs (A, B, D, E, and H). A classification can be more elaborate if it is based on several properties. For example, the properties *has legs* and *has wheels* divide the set into twoclasses: *{A,B,C,D,E,H}* and {F,G}.

DEFINITION 7: The set of all specifications, called *thefunction space* is a set of *all* the classifications of the functions. It is denoted by *T*

This definition does not modify the definition of specifications given in the introduction if the problem constraints are also reflected in the functions.

EXAMPLE: In principle, the function space for the chairs domains may contain 2^8 specifications, each being a different classification over the set of chairs. For example, the chairs that satisfy *support back, movable,* and *contemporary* is a classification that singles out chair D from the whole set of chairs. The number of specifications, however, is not accurate since some of the potential classifications (e.g., *standard* and *contemporary* or *movable, contemporary* and *stably support back*) do not contain any chairs.

DEFINITION 8: **The** set of all artifact descriptions, called the *attribute space* is a set of *all* the classifications of attributes. It is denoted by *%*

EXAMPLE: The attribute space for the chairs domain may contain 2* potential artifact descriptions. For example, the attributes *has legs* and *has lightweight,* designate the artifact descriptions of chairs F and G. Some of the classifications may be empty; however, it is easy to construct a new chair that will have the classification attributes. For example, the attributes *has wheels* and *is hanging* designate an empty artifact description; nevertheless, is is easy to hang a chair on wheels sliding on a rail that is attached to the ceiling.

3.3 GDT's axioms

GDT's axioms convey the assumptions of the theory about the nature of objects and their manipulation by humans. These axioms are the foundations of the theorems discussed later. The importance of the axioms lie in their coherent formulation. This allows the assessment of how close GDT approximates the real world.

AXIOM 1 **(Axiom** of **recognition):** Any entity can be *recognized* or *described* by its attributes and/or other abstract concepts.

EXAMPLE: Each of the chairs in Figure 2 can be easily singled out from the set of chairs by using one or more of its artifact description attributes. For example, chair A can be recognized as the only chair that have no vertical rotational dof and no wheels and chair C is the only chair that is not light-weight.

The only situation in which the axiom will fail is if two chairs have the same description. In this case, the two chairs cannot be differentiated. In reality, we often face decisions about which product to buy and our description (i.e., knowledge) is not sufficient to differentiate between the products. The key to remove this problem is to add attributes that differentiate between the artifacts. This entails that an entity may have a large, possibly infinite, number of attributes to enable its recognition. This, in turn, may require having infinite storage capacity and processing speed.

AXIOM 2 **(Axiom** of **correspondence):** The entity set and its representation has *one-to-one* correspondence.

EXAMPLE: If each of the chairs in Figure 2 is perceived as a real object (i.e., entity) and its description given in Tables 1 and 2 as the concept of entity, the axiom says that there is a one-to-one mapping between them.

The discussion on the previous axiom applies to the present axiom, therefore, to the representation of entities. It forces the use of an unbounded representation such as an infinitely long property-value list. Of course, this is practically impossible. Nevertheless, the theory assumes an infinite memory capacity and processing speed which overcome this difficulty.

AXIOM 3 **(Axiom of operation):** The set of all abstract concepts is a *topology* of the entity set.

A topology (S, *T)* is a mathematical entity consisting of a set *S* and the set T of subsets of *S* that satisfies the following properties:

(1) <£€ ***T* and *S* ∈ *T*,**

(2) for every *s*, s_2 € T, *s* n s_2 G T, and

(3) for every $1, s_2 € $T_y s_x$ Us_2 € *T*.

The fact that knowledge is a topology, influences both its structure (through property (1) of topology) and the possible operations on this knowledge (properties (2) and (3)).

EXAMPLE: The simplest topology over the set *S* of chairs is T = {<£, 5}. Another obvious topology is the power set of *S* (having 2^8 = 256 elements). As we see later, this topology is not too interesting. Another topology T can be is constructed such that {<£, *{A,H}* , *{B,C,D}* , {£,F,G} ,S} ⊆ T. The inclusion of these subsets of *S* may be caused by the need to *differentiate* between the entities in the different classes. To complete the topology, T must satisfy the three properties listed above. Therefore, T = {*, {A,//} , *{B,C,D}* , {£,F,G} , **{A,B,C,D,ff}** , *{B,C,D,E,F,G}* , {A,£,F,G,ff} ,S}.

3.4 Ideal knowledge

DEFINITION 9: *Ideal knowledge* is the one that knows *all* the entities and can describe each of them by abstract concepts without ambiguity.

This definition adds another constraint to the axioms: the recognition of entities should be without ambiguity. Furthermore, the recognition should be facilitated by the abstract concepts. This restrictive notion of recognition is one instance of the topological concept of separation which is formalized in the axiom of separation.

AXIOM 4 **(Axiom of Separation^[3]):** There exists a hierarchy of separation/recognition ability:

To: For **each pair *a*** ^ *b* in 5 there is *U* € *T* such that *a* £ *U* and *b* & *U* or vice versa.

Ti: Foreachpaira^6inStheieis£/,V € Tsuchthata G *Uandb* & *Umdb* 6 Vanda £ V.

72: (Hausdorff). Similar to *T*, but *U* n V = 0.

T3: (A regular 7i space). Satisfies Ti and for every closed set *A* C $\overline{S}$ (i.e., closure of *S)* and for every £ € S - A there exists a pair of disjoint open sets {/, V such that *b* £ *U* and A c *V (T3* is a generalization of *T2* when A is a single point).

T4: (A normal 7*i or a compact Hausdorff space). Satisfies *T* and for every pair of disjoint closed sets A,*B* C $\overline{S}$ there exists a pair of disjoint open sets £/, V € *T* such that *A C U* and B C V.

Ts: (A completely normal *T* space). Satisfies *T* and for every pair of closed sets A , B c $\overline{S}$ with $\overline{A}$ *D B* = *Af\\$\overline{B}$* = <$>, there exists a pair of disjoint open sets **(/ , V e T** such that *A C U* and B C V.

Metric space: There exists a metric on the space.

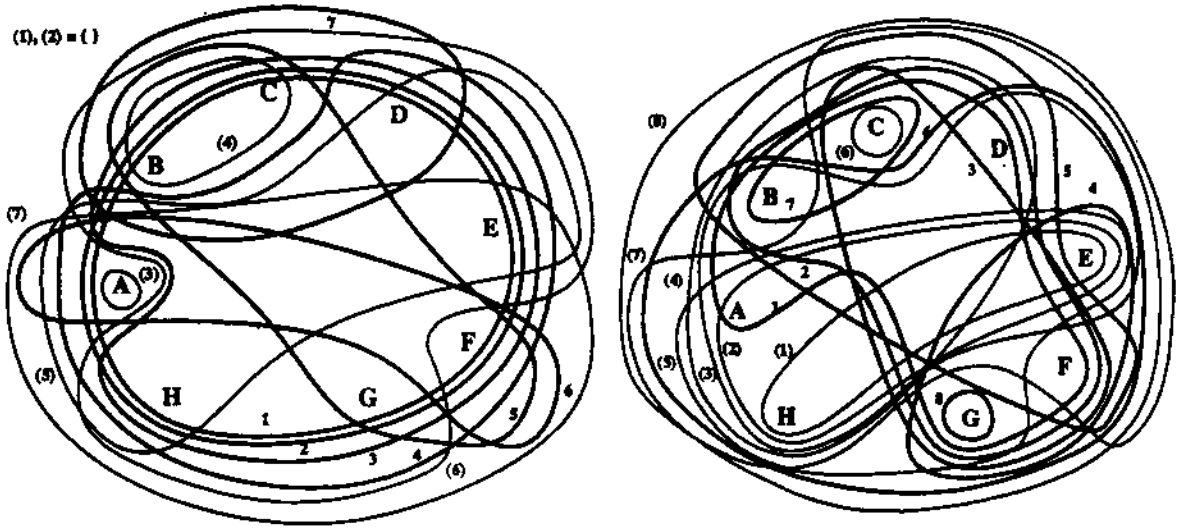
It can be shown that *metric* =» Ts => *T** => 73 =• 72 => 7i => 7o, and that none of these implications is reversible. Therefore, the type of separation defines an order on topological spaces.

EXAMPLE: Even though, the examples of topologies discussed before were correct, they do not constitute ideal knowledge because they do not allow the recognition of objects without ambiguity. To facilitate the recognition without ambiguity, each chair must be a member of T. This immediately implies T = 2^s. However, to be meaningful, we want the topology to have some meaning with respect to the domain. Therefore, the abstract concepts must be derived from properties describing chairs. Figure 4 illustrates the classes created from the properties in Table 2. Each line in the figure circles the chairs that satisfy the property number that is written along the line. A number in parenthesis denotes that the chairs have - as their value for this property. We immediately see that some entities, such as *D* is not recognized by an abstract concept. The problem can be partially remedied be introducing other observable attributes such as: *square seat* (differentiates *A), stiff seat* (differentiates //), *has a height adjusting screw* (differentiates F), *has a horizontal seat* (differentiates £), and *has two parts* (differentiates D)[4].

---

[3]**This symbol denotes that this axiom is not a part of GDT.**

[4]**In general, the process of verifying that each object or abstract concept can be discriminated without ambiguity from the remaining concepts can be performed using a learning program such as CN2 (Clark and Niblett, 1989). The learning program can generate a discriminating rule based on the attribute descriptions (e.g., attributes in Table 2) of the examples comprising the concepts.**



(a) Function "topology"

(b) Attribute "topology"

Figure 4: The function and attribute topologies

THEOREM 1: The ideal knowledge is a HausdorfTs space.

This theorem almost follows from the definition of ideal knowledge. Insisting that each entity can be described by abstract concepts without ambiguity only warrants a type-7*i separation and not necessarily type-T2. Insisting on a Hausdorff space is a strong interpretation of the word *ambiguity.' It remains to be detennined in future work whether a weaker interpretation can still lead to proving the same or similar theorems about design that GDT proves.

EXAMPLE: In the chairs domain, using Figure 4, entities *A* and *F* cannot be distinguished in the Hausdorff sense. They can, however, be distinguished in the *T* sense. For example, *{A,H,G,E}*, the abstract concept *does not have vertical rotational*do/—denoted by (5), and *{C,G,F}*> the abstract concept *has legs*—denoted by 3, differentiate between them but they are not mutually exclusive. On the other hand, entities *C* and *G* can be easily differentiated in the Hausdorff sense.

The next definition formalizes the intuitive definition of design specification given in the introduction. The definition is in correspondence with Definition 7.

DEFINITION 10: The design specification, T_s, designates the function of the required entity by using abstract concepts.

This definitions implies that T_s 6 *T*. Shortly, we see that a solution to the design problem is *s* 6 T_s *E *T*.

THEOREM 2: The specification can be described by the intersection of abstract concepts.

EXAMPLE: It is natural to describe the specification of an object by the intersection of abstract concepts since the specification describes functions that the desired chair must fulfill. Therefore, a specification that a chair must *revolve* and *be movable* is described easily (using Table 1) by {5,C,D,£,F,G} 0 {i4,D,£,f,G} = {D,£,F,G}. A solution to this design problem is any *s* € *{D,E,F,G}*. If the specification would insist on *ordinary design* {A, F}, the result would be *{D,E,F,G}n{A,F}={F}*

THEOREM 3: The set of design specifications 71 is a filter.

A filter allows access to an entity through a sequence of increasingly refined abstract concepts.

EXAMPLE: If the set of design specifications *T* contains *{A}*. Then from the definition of a filter, it contains all the supersets of {A}: *{A}t {A,F}9* {A,D,£,F}, {A,£,F,G,//}, {A,D,£,F,G,//}, and *S*.

THEOREM4: **T D 75 and T *D* T_X.**

This theorem establishes the relationships between the concepts of function and the ideal knowledge and the concepts of attributes and ideal knowledge.

DEFINITION 11: A design solution is an entity *s* that is *included in* its specification and carries out the necessary manufacturing information.

The notion of "carries out the necessary manufacturing information" is not defined. To illustrate the meaning, remember that ideal knowledge contains all the chairs that were manufactured and that will be manufactured. There will be similar chairs that will differ in the technology they are manufactured or in the tolerances on their dimensions. In order to differentiate between these chairs by some properties (i.e., abstract concepts) their description must contain all their dimensions, tolerances, and manufacturing techniques.

THEOREM 5: The entity concept in the ideal knowledge is a design solution.

This theorem states that *every* entity is a design solution because it satisfies some requirements and contains the necessary manufacturing information. The reason why each entity can have all the manufacturing knowledge necessary lies in the mathematical concept of neighborhood. A neighborhood is

any abstract **concept** that contains the entity. These neighborhoods can be accessed from the entity and as classification **based** on attributes such as tolerances and manufacturing technology, they contain the necessary manufacturing information.

EXAMPLE: In the specification *revolve, be movable,* and *ordinary design,* described before, *{F}* is the design solution. The neighborhoods of *F* that contain the manufacturing knowledge arc those elements in the attribute topology that contain *F* (see Figure 4(b)): {A,B,C,D,F,G}, *{B,C,D,F,G}, {C,F,G}>* {£,F,G}, *{B,C,D,F},* {A,fl,D,£,F,G,//}, {A,C,D,£,F,G,//}, and {A,£,C,D,£,F,//}. Of course, these neighborhoods correspond to the attribute description of chair *F* that appears in Table 2. The translation of these attributes to manufacturing information is simple: the chair will be manufactured with a seat, back support, legs, wheels, etc.

THEOREM 6: The entity concept in the attribute space $(S, \bar{T}o)$[5] is a design solution. Each of the attributes can be perceived as a manufacturing information.

The attribute space is the one that is created by the observable properties. These are the ones that are needed to manufacture an entity. Therefore en entity in the attribute space is a design solution. The next definition formalizes the notion of design solution described in the introduction.

THEOREM 7: The design solution is represented by the intersection of classes of *S* that belong to the attribute space $\bar{T}o$.

EXAMPLE: From Figure 4(b) it is obvious that *C* and G which are isolated chairs are design solutions.

The next definition formalizes the intuitive definition of design given in the introduction. Its two subsequent theorems establish the nature of design in the state of ideal knowledge. Note that in the ideal state design is similar to synthesis.

DEFINITION 12: Design process is the designation of a domain in the attribute space $(5, \bar{T}o)$ which corresponds to a domain specifying the specification in (S, T).

THEOREM 8: If the function space is a subspace of the attribute space, then design is complete when the specification is provided.

[5](5, T) is the mathematical notation of a space, we sometimes use T to denote a space as well.

EXAMPLE: Let the specification be to design a chair that *stably supports the back* and that *is movable.* Specification 1 (a chair that *stably supports the back)* is {A,fl,C,£,F,G}. The addition of specification 2 (a *movable* chair), *{C,D,E,F,G}>* generates the specification *{C,E,F,G}.* The only way that a design solution exists is that there is a class in the attribute space that is a subset of {C\£,F,G}. In the chairs domain, {C},{G},{C,F,G}, and {£,F,G} are subsets of the specification and therefore may be design solutions. Unfortunately, both *{C,F,G}* and {£,F,G} do not contain all the necessary manufacturing information. For example, *{C$_y$F,G}* does not specify whether it should have wheels and {£, *F, G*} does not specify whether it should have legs.

THEOREM 9: In the ideal knowledge, design is completed immediately when the specification is described.

EXAMPLE: Let the specification be to design a chair that *stably supports the back, is movable,* and *is standard.* Specification 1 (a chair that *stably supports the back)* is *{A,B$_y$C,E,F,G}.* The addition of specification 2 *(a movable* chair), *{C,D,E,F,G},* generates the specification *{C,E,F,G}.* The further addition of specification 3 *(standard),* {A,F}, leads to the combined specification *{F}.* At this stage all the specification have been described and the resulting design solution is *{F}.*

The next definition refines the previous Definition 12.

DEFINITION 13: Design is a mapping between the function space to the attribute space[6].

THEOREM 10: The identity mapping between the attribute space to the function space is continuous.

The interpretation of the theorem is that Tō 3 71. Intuitively is suggests that any class in the function space is equal or contain several classes from the attribute space; therefore, if a specification is determined there will always be at least one candidate solution.

THEOREM 11: If two design solutions can be discriminated functionally, then Tō C *T\.*

This theorem, however, does not hold if design is possible unless both topologies are equal, because then, from Theorem 10, T_o *D T\.*

Summary of ideal knowledge. The state of ideal knowledge is characterized by the ability to separate between entities. This separation may require the use of infinite descriptions of entities. The separation

[6]Remember that the function and attribute spaces are topologies over the same set of objects.

between entities and the requirement that the knowledge structure be a topology guarantee that design would terminate with a solution after a specification is given. The operation on large description of entities by the set operations that manipulate the topology may require infinite memory capacity and processing speed.

To summarize, GDT-IDEAL, denoting the state of ideal knowledge, restricts the nature of knowledge to have two perfect properties, but can then guarantee the termination of design. Real design never has these properties and therefore, its termination cannot be guaranteed. Nevertheless, several design strategies, and additional assumptions can guarantee design in a less ideal state called GDT-REAL. These issues are discussed next.

3.5 Real knowledge

The real knowledge cannot be structured as a perfect topology and cannot be manipulated by infinite resources. These discrepancies from the state of ideal knowledge require two important modifications to the theory. First, only finite descriptions of entities can be manipulated. Second, instead of assuring the successful termination of design, alternative models of design must be devised to allow solving design problems. The alternatives rely on the identification of a mapping between the function and the attribute topologies which are now imperfect.

For example, designing a chair that will *seat* and be *of contemporary design, stably support back* and *movable* is impossibly. The intersection of the classes corresponding to the requested properties does not contain any chair although the existence of such chair is guaranteed by GDT-IDEAL. The design can be accomplished by the addition of a *stopper* to chair *D* to make it *stable.* This however, involves enhancing the topologies which is a creative action.

The remainder of this section reviews the definitions, assumptions, and theories of the extension of GDT to real knowledge denoted by GDT-REAL. Since, most of the effort in this part of GDT goes into proving similar theories about design as in the state of ideal knowledge, examples will be limited to the distinctions between GDT-IDEAL and GDT-REAL.

DEFINITION 14: A *physical law* is a description about the relationship between object properties and its environment.

EXAMPLE: Physical laws include: gravity which establishes the *lightweight* property, vision which establishes all the properties that can be observed, etc.

DEFINITION 15: An *attribute* is a physical quantity which is identifiable using a set of *finite* number of physical laws

This definition is in correspondence with Definition 3 except for the addition of the *finite* adjective. Insisting on a finite number of laws has an important implication that is later given a precise meaning in Hypothesis 1. This hypothesis is one of the core assumptions of GDT-REAL.

EXAMPLE: All the observable properties in the chairs domain can be identified by physical laws such as gravity or vision, therefore they are attributes.

DEFINITION 16: A concept of physical law is an abstract concept. It is formed if entities are classified based on physical manifestations due to physical laws.

If each physical law corresponds to one and only one observable property, the topology of physical law is equal to that of the attribute space, otherwise it may be slightly different.

GDT now makes a distinction between 5, the set of entity concepts, and $\tilde{5}$ which is the set of entities that do not contradict physical laws. This distinction is not too important since *S* is the representation of existing entities via one-to-one mapping. Therefore, *S* contains only feasible objects. The distinctions between the two sets is ignored henceafter.

DEFINITION 17: A feasible object is an object that does not contradict physical laws.

EXAMPLE: A feasible object is an object that can be realized. Since all the chairs are examples of existing designs, they are all feasible.

THEOREM 12: The topology of physical law, T_p, on $\tilde{S}$ is a *subspace* of $T\tilde{o}$.

The fact that T_p is a subspace of $T\tilde{o}$ guarantees that anything that can be expressed by physical laws will be realizable as an artifact. Therefore, if specifications can be described by physical laws, design is possible.

HYPOTHESIS 1: The real knowledge is the set of feasible entity concepts which are made compact by coverings selected from the physical law topology.

EXAMPLE: Any finite space (e.g., the chairs domain) or a space with a finite number of open sets is compact.

Conceptually, this hypothesis says that the description of entities is restricted to a finite number of properties. Formally, insisting on a compact space is a very restrictive hypothesis. It demands the existence of a finite subcover for each cover of the space. From the formal perspective, this hypothesis is as important foundation of GDT similar to the assumption that the structure of knowledge is topology (Axiom 3). It almost immediately determines all the properties discussed henceforth.

To illustrate, compactness guarantees the feasibility of a specification (e.g., the intersection of the classes representing the specification properties is not empty) if they can be represented as closed sets and if the specification is feasible for any finite collection of properties. This is formalized in the following theorem (Christenson and Voxman, 1977; p. 67):

THEOREM 13 (t): A space *S* is compact if and only if for each collection *C* = *{C* A € *A}* of closed subsets of 5 with the finite intersection property, HCA,A€A *jt* <*t*>

Another immediate result is that the compactness property with the Hausdorff property of real knowledge, determines the metrizability of real knowledge. This implication is further discussed later.

The following three theorems are stated for completeness but are not discussed further.

THEOREM 14: The topology of To on the set of feasible concept *S* is a compact Hausdorff space.

THEOREM 15: The real knowledge is second countable (i.e., has a countable basis).

THEOREM 16: The real knowledge is a closed subset of the (ideal) set of entity concept S.

THEOREM 17: If a continuous function/ *:S->R* exists in the real knowledge, this function has the maximum and minimum value.

This important theorem says that continuous functions have finite values ranging between two limits. This property is necessary when dealing with functions that represent properties of physical objects.

The next theorem is stated for completeness since it is used in proving later theorems. Intuitively, a Lindelöf space is a less restrictive notion of a compact Hausdorff space.

THEOREM 18: The real knowledge is a Lindelöf space.

DEFINITION 18: Let T be the topology of abstract concept and *A* a countable set. *Feasible design specification,* $T = n_{XeA}T_x(T_x$ e T, $T_x \pm <t>\backslash$ is defined by the following condition: nr_A *ft* <*t*>.

This definition adds to Theorem 2 the restriction on a *countable* intersection of abstract concepts. This guarantees that the specification will be feasible (i.e., contain candidate designs).

THEOREM 19: Any feasible specification in the real knowledge has a cluster point.

This conclusion is equivalent to Hypothesis 1 (see (Schubert, 1968); p. 69). The set of candidate designs will be a set of entities that also contains the cluster point. The next theorem shows how a converging subsequence can be built that will arrive at a single solution. The proof postulates the existence of a function that can select one entity from a set. The existence of such function is guaranteed by the Axiom of selection. In design terminology, it corresponds to the style a designer has.

THEOREM 20: In the real knowledge, it is possible to make a converging subsequence from any design specification and to find out the design solution for the specification.

This and the next theorems restate Theorem 9 for the state of real knowledge.

THEOREM 21: In the real knowledge the design specification converges to one point, if it is possible to get a directed sequence of points from the specification.

The following definitions and theorems deal with the important concept of models. The key idea is that models are described by finite number of attributes. This is in correspondence with Hypothesis 1. Intuitively, models are some abstraction of entities that focus on few properties. In many cases, models are then subjected to some analysis based on a theory to find behaviors that are attributed to the original entity.

DEFINITION 19: A metamodel *MA* is defined as *r\\£AM\,(M* € *%\ A* is finite.

EXAMPLE: Modeling the chairs as two dimensional entities and the application of statics laws lead to estimating the behavior of chairs *A, D,* £, and *Has not stably support back* and chairs *B,* C, F, and *G* as *stable.*

DEFINITION 20: The *metamodel set, M,* is the set of metamodels that are formed by finite intersections.

THEOREM 22: The metamodel set is a topology of the real knowledge.

THEOREM 23: In real knowledge, the metamodel set is a topology *weaker* than the attribute topology.

The term *weaker* means that the metamodel set can only approximate the attribute topology.

THEOREM 24: In the real knowledge, the *necessary* condition for designing is if the topology of the metamodel set is stronger than the topology of the function space.

Since *TQ D M,* then if *M D T* then *% D M D T*. This implies *TQ D T* which is the condition for the ability to design.

THEOREM 25: If designing is possible, the identity mapping from *%* to *M* is continuous.

This theorem is similar to Theorem 10 in GDT-IDEAL.

THEOREM 26: If the identity mapping from *M* to *T* is continuous, the identity mapping from 7ŏ to *T* is continuous.

THEOREM 27: If we evolve a metamodel, we get an entity concept as the limit of the evolution.

Theorem 21 is used to prove this theorem, except that here, the limit is an element of *M.* Therefore, it may only be an approximation of the true design solution.

DEFINITION 21: *A function* of an entity is a physical phenomenon caused by the physical laws governing the situation.

This is the intuitive notion of a behavior of an object when exposed to a physical situation, for example, push a chair and see whether it is stable. This definition slightly differs with Definition 4. This definition is directed towards defining function by attributes and concept by a finite number of physical laws (e.g., length, weight, or color).

DEFINITION 22: *A function element* is a metamodel $n_{X£A}M$ A, (Af $_x$ € *%, A* is a finite set), such that VM_A,Af_A € T_p{\ € *A)*

The next theorem is the one corresponding to Theorem 9 in GDT-IDEAL. It does not guarantee arriving at a solution when the specification is described, but guarantees the finding of an approximate design solution.

THEOREM 28: If we choose function elements as the metamodel, design specification is described by the topology of metamodel, and there exists a design solution that is an element of this metamodel.

THEOREM 29: **The** real knowledge is a normal space.

This theorem says that the topology of real knowledge is finer than what originally perceived (i.e., a retype space instead of a ^-type space). Shortly, it is proved that it is even finer (i.e., it is a metric space).

THEOREM 30: In the real knowledge there exists a distance between two different entities.

The topology of the real knowledge is "fine" enough to allow the calculation of a metric. This is one of the most important results of GDT. The distance in a metric space supports incremental refinement and easy redesign. A simple example of the benefits from properties derived from a metric (continuity, convergence, etc.) was discussed in Section 3.1. Such a fine topology supports the construction of knowledge by metrization (Taura et al., 1989). The ability to create a metric, however, can be derived from less restrictive assumptions. The least committing theorem on metrizability is:

THEOREM 31 **(Nagata (1950) and Smirnov (1953),t):** A space *S* is metrizable if and only if 5 is 7_3 and has a a-locally finite basis.

The next theorem restates that if a space is metric than the distance measure can be used to assign values to each of the attributes describing an entity.

THEOREM 32: In the real knowledge an attribute has a value.

The following definition and theorem states that the characteristics of real knowledge do not allow a specification that will result in a single design. Therefore, the candidates designs have behavior that is not specified or unexpected.

DEFINITION 23: When a design solution is materialized, although it is exposed to a specific situation, it may have behaviors different from the specifications. These behaviors are called *unexpected functions.*

THEOREM 33: In the real knowledge the design solution has unexpected functions.

Summary of real knowledge. The state of real knowledge is an adaptation of the concept of ideal knowledge to the real world. Entities are described by a finite number of attributes that can be measured

by physical laws. Therefore, these descriptions can be manipulated in finite time and require finite storage capacity. The main assumption of the topological nature of knowledge remains intact.

Design in the state of real knowledge requires the ability to continuously model the designed artifact until it evolves into a candidate that satisfies the specification. A model is an abstraction of the actual entity and serves as a focus for the current stage of the design. The use of models guarantees an approximate solution in the state of real knowledge.

Essentially, GDT-REAL is similar to GDT-IDEAL with the addition of the restriction to finite properties, and the introduction of models as mediators from the specification topology to the attribute topology.

4 Discussion

A strong motivation of GDT is to provide a better prescription for building design support systems. The contribution of the theory can be assessed by its influence on the design of future design support systems. Three issues establish this influence. The first issue is concerned with the scope of the theory, namely, how many domain can it describe?. The second issue is concerned with a set of general guidelines for building design systems that GDT offers. The third issue is concerned with the possibility of actually implementing a system that operates similar to GDT. This section elaborate these issues.

4.1 The scope of GDT

General Design Theory proves strong theorems about design (e.g., Theorems 9 in GDT-IDEAL and 28 in GDT-REAL). It arrives at these theorems by imposing restrictions on design knowledge. The requirements on infinite processing speed and memory capacity laid in GDT-IDEAL are relaxed in GDT-REAL at the expense of obtaining solutions that are close approximations of design solutions. Since the latter is sufficiently detailed to allow manufacturing, the only remaining restriction is the representation of design knowledge.

The restrictions on knowledge representation emerge from the assumptions of GDT. Some of these assumptions can be relaxed. First, in GDT-IDEAL, the definition of ideal knowledge can be interpret as a less restrictive notion of the Axiom of separation. Second, in GDT-REAL, the hypothesis that real knowledge is a compact space is also restrictive. In particular, since the important results of GDT-REAL are proved using the Lindelöf space property, which insist on a countable and not finite subcovering of any existing cover, there is no need to assume compactness. Third, the desire to prove metrizability of

real **knowledge** is **not** necessary. In part II of the study we show that without the use of a metric space, hierarchical **knowledge** structures can be generated and support effective design.

These three potential relaxations can still maintain sufficient ground for proving the same theorems that GDT proves. Moreover, we are interested in considerably less restrictive assumptions that can support less promising but useful and more realistic predictions about designability. Such relaxations will be discussed in Part II of the paper.

To summarize, the scope of GDT can be broaden by insisting on less restrictive assumptions. The extend to which these assumptions can be relaxed depends on the ability to use them effectively in proving results similar to those currently obtained.

4.2 Guidelines for building design systems

Even if GDT has a restricted scope, it is still useful for guiding the construction of design support systems. This section draws on the papers describing GDT and others, by Tomiyama, Yoshikawa, and their colleagues, discussing related issues. The guidelines are organized along the representation and process aspects.

Representation. Two potential representations exist for representing objects: extensional and intentional. In the extensional representation, an attribute is expressed as the set of objects having this attribute. This representation of the chairs domain appears in Figure 4. In the intentional representation objects are described by the set of attributes characterizing them. Tables 1 and 2 describe the chairs domain intentionally. Axioms 1 supports the use of intentional representation of entities, and Axiom 3 and the way design process is executed (i.e., by the intersection of abstract concepts) demand the use of an extensional representation of abstract concepts[7].

Each of the two representations have advantages and disadvantages which can be summarized as follows (Tomiyama and Yoshikawa, 1986; Tomiyama and ten Hagen, 1987). An extensional description of an object can be easily modified but may hardly convey the complete meaning of the object description. In contrast, an intentional description of an object is hard to modify but its meaning can be easily understood. These differences, in addition to the ease of computer implementation of intentional descriptions, demand

[7]Studies on GDT sometimes disagree on this issue. For example, Tomiyama and Yoshikawa (1986) say that (p. 5) "It is that the description method of an entity concept must be extensional or denotative, but not intentional nor connotativc." (boldface in the original manuscript). In contrast, Kunimatani and Yoshikawa (1987) say that (p. 724) "Axiom 1 guarantees the possibility of recognizing an entity by its connotation instead of its denotation/'

the use of **both representations** in design support systems. This is in complete agreement with our understanding of **GDT.**

Figure 5 summarizes a discussion in (Tomiyama and Yoshikawa, 1985) suggesting that extensional description are best at the beginning of the conceptual design phase since they facilitate easy transformation from functions to artifact description. In the detail design phase, objects are detailed with additional attributes, thereby preferring an intentional representation.

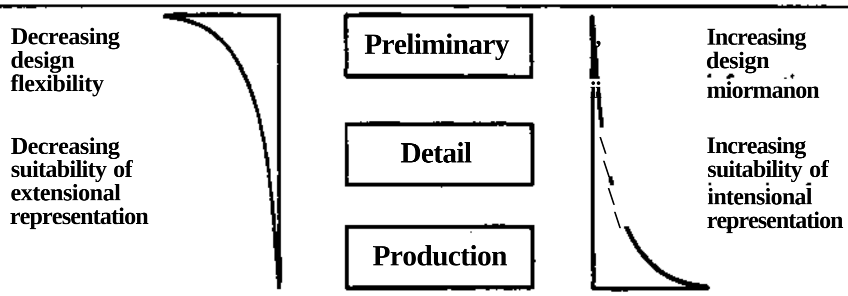


Figure 5: Representation for different design phases.

The second aspect of representation of knowledge is the overall organization of concepts and entities. GDT is based on a topological structure of the universe of entities. Nevertheless, it recognizes that in real design topological structure does not exists. Therefore, Hypothesis 3, discussed shortly, postulates that hierarchical knowledge structure is the representation used by human designers.

Process. In his original English paper, Yoshikawa addresses the issue of real versus ideal knowledge. Real knowledge does not lead to a design solution when the specification is described. Yoshikawa suggests to remedy the shortcomings of real knowledge by providing a mapping between the function and the attribute topologies. This mapping can be generated by one of the following methods:

(1) *Correspondence methods.* These methods provide limited, but direct mapping between the function and the attribute topologies.

  (1) *Catalogue.* This method can be realized in an information retrieval system that provides a solution from the best available solution. In the chair domain, it means selecting an existing chair that can best satisfy the requirements. In real design, rarely will a catalogue contain exactly the desired item.

  (2) *Calculation.* In this method, numerical procedures are used to derive a design solution. The calculations are expected to be in the form where attribute properties are calculated from functional properties. This method is applicable to simple

design problems were a mathematical model can be formulated and solved.

(3) *Production.* In this method, functions are transferred into attributes by production rules. Currently, this method is also applicable to simple design problems.

(2) *Convergence methods.* In this process functions are added incrementally and the design gradually converges to having the desired behavior.

When applicable, the correspondence models should be employed since they provide direct mapping between function to attribute description. In general, however, the convergence process is the most promising. This is due to its evolutionary nature which does not require to arrive immediately at the final description of the design solution.

To facilitate the convergence method, two hypotheses are forwarded that support some of the structure of topologies in real design, thereby maintaining some of the properties discussed in the preview section (see (Yoshikawa, 1981); p. 49-50). The first hypothesis tries to conserve some of the benefits from continuity.

HYPOTHESIS 2: The more similarities of function there are between two entities, the more are the similarities of attribute between them. The converse also holds.

The second hypothesis attempts to maintain some of the benefits from infinite resources that can intersect arbitrary sets. The constraint on the structure of knowledge supports its effective management and use.

HYPOTHESIS 3: The entity concept in the designer has a hierarchical structure.

GDT-REAL presents the concept of models as mediating from the specification to the attribute topologies. When using the convergence method, models must be used for supporting the incremental modification of the design toward the solution. Theorems 27 and 28 guarantee arriving at an approximate solution when using this correspondence model.

Figure 6 illustrates the concept of design as a converging process using the chairs domain. The figure shows the addition of specification requirements and the application of models for maintaining candidates that satisfy all the previous requirements. The process terminates with an acceptable design. It is important to note that the order of adding functions is very important for the efficiency of the process. The length difference between a good and a bad ordering can be between *log* and *linear* of the number of chairs.

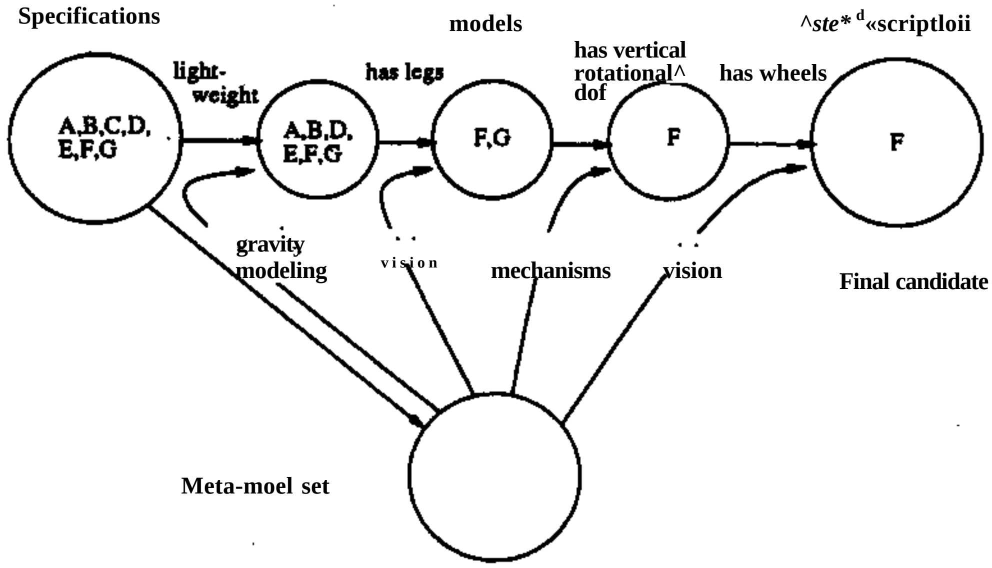


Figure 6: Design as a converging process

4.3 Implementation of GDT

This section deals with the implementation of the guidelines discussed before. A complete implementation must follow as many guidelines as possible and remain coherent. In particular, the implementation must answer two fundamental questions: how is the entities representation (i.e., topology) generated and how is the process (i.e., the mapping from the function to the attribute topologies) represented? These questions remain open in the formalization of GDT as summarized in Figure 7.

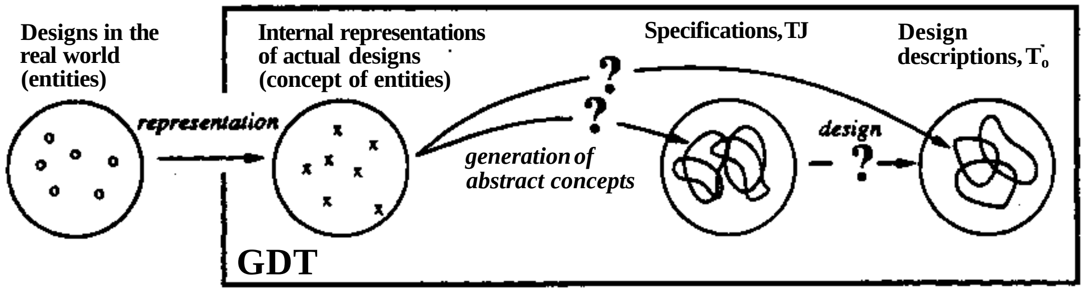


Figure 7: Learning: The missing part in GDT

Representation. The interleaved extensional and intentional representation discussed before was implemented in IDDL (Integrated Data Description Language) (Tomiyama and ten Hagen, 1987; Vcih, 1987). Among other things, IDDL supports extensional and intentional descriptions, and can describe objects and processes, metamodels and models. IDDL can also support the construction of a hierarchical structure. Overall, IDDL is a *manual* method for coding knowledge and entities based on the guidelines set by GDT. Unfortunately, IDDL provides no answer to the question of how topologies are generated.

A method for constructing knowledge structure was experimented in the context of GDT. It involves the creation of a metric space by the use of a clustering technique (Taura et al., 1989). In this approach, a metric is used to represent the distance between the function of or behavior manifested by designs. The function or behavior is manually extracted from the artifacts* descriptions. A similarity function is used to measure the distance between any two artifacts, arranged in a similarity matrix (see Figure 8). Several important dimensions Ai,..., A/ are extracted by performing a principal coordinate analysis over the matrix. The analysis reduces the dimensionality of the description space, which in turn allows a simple clustering. Each cluster is then identified by generalizing the design descriptions of its elements. This description is then used in new situations to construct new objects to satisfy a given functionality expressed by the principal coordinates. This approach generates a single-layer classification and not a hierarchical structure. Therefore the knowledge generated is computationally inefficient for the retrieval of designs that are similar to the new situation.

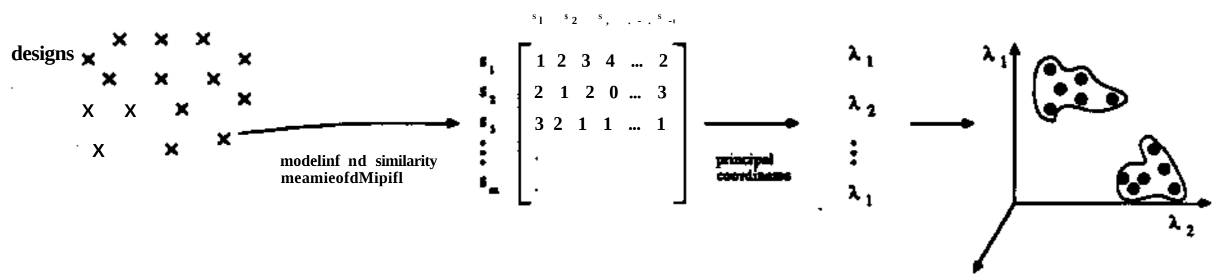


Figure 8: Creation of a metric space

To summarize, studies related to GDT have provided guidelines and tools for the representation of knowledge. They have not addressed the issue of how to generate efficient, hierarchical knowledge structures that support extensional as well as intentional descriptions. In particular, they have not discussed how this can be done automatically and incrementally in response to changing technology.

Process. As mentioned before, IDDL is also capable of representing design processes. In particular, it supports the representation of metamodels, models, and the evolutionary control of design required by the correspondence model.

IDDL can therefore be used to manually describe design processes. However, the mapping between the function and the attribute topologies cannot be easily manually coded. Therefore, IDDL does not provide an answer to the question of how is the mapping generated automatically and incrementally in response to experience.

5 Summary

This study introduced the scientific method as a research activity grounded in a recurring cycle of theory generation, experimental testing, and theory refinement. The goal of of this study is to demonstrate that the scientific method of conducting research is the one to adopt. This demonstration is based on a detailed case study.

This paper, which is part I of the study, critically reviewed GDT—a well developed theory of design. It analyzed the assumptions made by GDT, the theorems it proves, and the ramification it has on building design systems. The crucial points of GDT were highlighted and analyzed. In light of the restricted scope of GDT, the review suggested some relaxations of assumptions that may maintain the same predictions about the convergence of design processes. The analysis of this paper and its conclusions can be viewed as completing one cycle of the scientific method while remaining in the mathematical framework of the theory.

The state of GDT implementation in a design support system was also discussed. Currently, there is no system that can explain how topologies are generated and how the mapping between the function and the attribute topologies is formulated from experience. Since GDT also aims at explaining human design processes (Yoshikawa, 1981; p. 35), these processes must be accounted for. In the absence of these abilities, experimental testing of the theory remains a hard task of coding knowledge and design processes in the right way and using them to design. Such experiments have not been conducted to date.

Beside laying the ground for the second part of the study, this part contributes to a better understanding of GDT and to its better assimilation and future development. Part II of the study discusses the design, implementation, and testing of an experimental design system, called BRIDGER, based on the foundation of GDT. BRIDGER can serve as an experimental support for GDT. The results of the study provide insight on how to modify BRIDGER to enhance its performance and on how to refine GDT to broaden its scope to more realistic design scenarios. These results show that subscribing to the scientific method results in fast and focussed progress.

Acknowledgments

This work has supported in part by the Engineering Design Research Center, a National Science Foundation Engineering Research Center.

References

Antonsson, E. K. (1987). Development and testing of hypotheses in engineering design research. *ASME Journal of Mechanisms, Transmissions, and Automation in Design,* 109:153-154.

Christenson, C. O. and Voxman, W. L. (1977). *Aspects of Topology.* Marcel Dekker, New York.

Clark, T. and Niblett, P. (1989). The CN2 induction algorithm. *Machine Learning,* 3(4):261-283.

Dixon, J. R. (1987). On research methodology towards a scientific theory of engineering design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* 1(3): 145-157.

Eastman, C. M. (1991). Use of data modeling in the conceptual structuring of design problems. In Schmitt, G. N., editor, *CAAD Futures '91: Proceedings,* pages 207-223, Zürich. Department of Architecture, Swiss Federal Institute of Technology.

Eder, W. E. (1987). Theory of technical systems - prerequisite to design theory. In Eder, W. E., editor, *Proceedings of The 1987 International Conference on Engineering Design, ICED-87,* pages 103-113, Boston, MA. American Society of Mechanical Engineers.

Filippone, S. F. (1989). Using Taguchi methods to apply the axioms of design. *Robotics & Computer-Integrated Manufacturing,* 6(2): 133-142.

Hongo, K. (1985). On the significance of the theory of design. In Yoshikawa, H., editor, *Design and Synthesis, Proceedings of the International Symposium on Design and Synthesis, Tokyo, Japan,* pages 169-174. North-Holland, Amsterdam.

Hubka, V. and Eder, W. E. (1990). Design knowledge: Theory in support of practice. *Journal of Engineering Design,* l(l):97-108.

Kurumatani, K. and Yoshikawa, H. (1987). Representation of design knowledge based on general design theory. In Eder, W. E., editor, *Proceedings of The 1987 International Conference on Engineering Design, ICED-87,* pages 723-730, Boston, MA. American Society of Mechanical Engineers.

Nagata, J. (1950). On a necessary and sufficient condition of metrizability. *Journal Institute of Polytechnics,* 1:93-100.

Popper, K. (1965). *Conjectures and Refutations: The Growth of Scientific Knowledge.* Harper and Row, New York.

Reich, Y. (1990a). Converging to "Ideal" design knowledge by learning. In Fitzhom, P. A., editor, *Proceedings of The First International Workshop on Formal Methods in Engineering Design,* pages 330-349, Colorado Springs, Colorado.

Reich, Y. (1990b). Design knowledge acquisition: Task analysis and a partial implementation. *Knowledge Acquisition,* in Press.

Reich, Y. (1991a). *Building and Improving Design Systems: A Machine Learning Approach.* PhD thesis, Department of Civil Engineering, Carnegie Mellon University, Pittsburgh, PA. Available as Technical Report EDRC 02-16-91.

Reich, Y. (1991b). Design theory and practice II: A comparison between a theory of design and an experimental design system. Technical Report EDRC 12-46-91, Engineering Design Research Center, Carnegie Mellon University, Pittsburgh, PA.

Rohatyfiski, R. (1983). On a classification of design problems relevant to design methods and strategy. *Schriftenreihe WDK10,*2(508).

Schubert, H. (1968). *Topology.* MacDonald Technical & Scientific, London.

Smirnov, Y. M. (1953). On metrization of topological spaces. *American Mathematical Society Translation, Series* 7, 8:62-77.

Suh, N. P. (1984). Development of the science base for the manufacturing field through the axiomatic approach. *Robotics & Computer-Integrated Manufacturing,* 1(3/4):397^115.

Suh, N. P., Bell, A. C, and Gossard, D. C. (1978). On the axiomatic approach to manufacturing and manufacturing systems. *Journal of Engineering for Industry, Transaction of the ASME%* 100:127-130.

Sutherland, W. A. (1975). *Introduction to Metric and Topological Spaces.* Oxford University Press, Oxford, UK.

Taguchi, G. (1980). *Off-Line Quality Control.* Central Japan Quality Control Association, Nagoya.

Taguchi, G. (1986). *On-Line Quality Control.* Japanese Standards Associations, Tokyo.

Taura, T, Kubo, S., and Yoshikawa, H. (1989). Generation of design solutions by a metric space. In *Proceedings of the Third IF IP W.G 52 Workshop on Intelligent CAD, Osaka, Japan.*

Tomiyama, T., Kiriyama, T., Takeda, H., Xue, D., and Yoshikawa, H. (1989). Metamodel: A key to intelligent CAD systems. *Research in Engineering Design,* 1(1): 19-34.

Tomiyama, T. and ten Hagen, P. J. W. (1987). Representing knowledge in two distinct descriptions: Extensional vs. intentional. Technical Report CS-R8728, Centre for Mathematics and Computer Science, Amsterdam.

Tomiyama, T. and Yoshikawa, H. (1985). Knowledge engineering and CAD. In Yoshikawa, H., editor, *Design and Synthesis, Proceedings of the International Symposium on Design and Synthesis,* pages 3-8. North-Holland, Tokyo, Japan.

Tomiyama, T. and Yoshikawa, H. (1986). Extended general design theory. Technical Report CS-R8604, Centre for Mathematics and Computer Science, Amsterdam.

6ech, E. (1966). *Topological Spaces.* John Wiley & Sons, London.

Veth, B. (1987). An integrated data description language for coding design knowledge. In *Intelligent CAD Systems /,* pages 295-313. Springer-Verlag, Berlin.

Warfield, J. N. (1990). *A Science of Generic Design.* Intersystems Publications, Salinas, CA.

Yoshikawa, H. (1981). General Design Theory and a CAD system. In Sata, T. and Warman, E., editors, *Man-Machine Communication In CAD/CAM', Proceedings of the IFIP WG5.2-53 Working Conference,* pages 35-57. North-Holland, Amsterdam.

Appendix. A glossary of mathematical concepts

This appendix defines the set theory and topological terms used in this study. The definitions listed were compiled from (Čech, 1966; Christenson and Voxman, 1977; Schubert, 1968; Sutherland, 1975) and appear in alphabetical, rather than dependency, order. The terminology used in this study is at the introductory level of topology.

Axiom of selection/choice (simultaneous selection): If there exists a collection of disjoint sets, then there exists a set that has precisely one element from each of the sets in the collection.

Basis: If (5, T) is a topological space, then a basis for *T* is a subcollection *B* of T such that if *s* G *S* and *U* is an open set containing *s,* then there is a set *V e B* such that *x* 6 *V* c *U.*

Cauchy sequence: A sequence (s_n) in a metric space *S* with a metric *d* is a *Cauchy sequence* if given $e > 0$, there exists N such that $d(s_n, s_m) < e$ for all $n,m \geq N$.

Closed set: A subset of a topological space is closed if its complement is open.

Closure: Let (5, T) be a topological space and *A* c 5. The closure of *A* in *S* is defined as *f){C\C* is closed in *S* and *A* c *C}* and denoted by $\overline{A}$.

Cluster point: A point *xeX* belonging to the closure of *X* - *(x)* is a cluster point. Intuitively, a cluster point is a point where many other points accumulate and converge to that point.

Compact: A topological space is compact if every open cover has a finite (not just countable!) cover.

Continuous mapping: Let *(A\,T\)*, (A_2,T_2) be two topological spaces, a mapping/ : *A* -> A_2 is continuous if for all *U* € T_2 => $f'^{l}(U)$ *e* *T*.

Convergence of a sequence: A sequence $\{s_n)$ in a topological space 5 *converges* to a point *s* € *S* if and only if for every neighborhood *U* of *s* there is a positive integer *Nu* such that *Si* *e* *U* whenever *i* > *Nu.*

Countable basis: A basis is countable if it has a countable number of sets.

Cover: A cover of a set 5 is a collection of sets whose union is a superset of *S.*

Directed set: A set *S* with a partial order ^ is called a *directed set* if for each *s^Sj* € 5, there is *Sk* G *S* such that *Si* •< *Sk* and *Sj* < *Sk-* A *directed sequence* is the same except that now also *i* < *k andj* ≤ *k.*

Family: A family *A* of elements of 5 is a set A_9 a mapping *a :* A -+ 5, and the subset <*r(A)* of 5.

Filter: A filter of *S* is a collection *T* of subsets of 5 that has the following properties: (1) *4>* £ *T,* (2) if *A* *e* *T*> andA c *B* C 5,thenfi C *T,* and (3) if A, *B* 6 *TthcnADB* € *T.*

Finite intersection property: A collection of sets *C* = {CA|A € A} has the *finite intersection property* if and only if for each nonempty finite subset *N* C *A, HCn^eN* 7* <*t*>

Hausdorff space: A Hausdorflf space is a space S that satisfies the condition: any two distinct entities in S can be surrounded by disjoint neighborhoods.

Homeomorphism: If (Si,7i) and (52,72) are topological spaces, then a mapping/ *: S* -* S_2 is called a *homeomorphism* if and only if/ is invertible and both/ and/ $^{\sim 1}$ are continuous.

Lindelöf space: A Lindelöf space is a space that satisfies the condition that every open covering of the space has a countable subcovering.

Locally finite: A family *A* = $\{A_A\}$,\€.i of subsets of a topological space 5, is *locally finite* if for every *s* € *S* there is a neighborhood *U(s)* such that *U(s)* n *A* ^ <*t*> for at most a finite number of A.

(7-locally finite: A cover U of a space is a-locally finite if and only if U can be expressed as the union of a **countable** collection of families, each of which is locally finite.

Metric space: A set S is called a *metric space* if with every pair of points x,y € 5, there exists a non-negative real number $d(x,y)$ that satisfies:

(1) If $d(x_y y) = 0$ then $x = y$ and $d(x,x) = 0$ always holds.

(2) For any pair of points $x, y> d(x,y) = d(y,x)$.

(3) For any three points $x_n y_\%$ and z, $d(x,z) \leq d(x,y) + d(y_y z)$.

Neighborhood: If (5, T) is a topological space then the neighborhood of s € S is any of the sets U € T such that s € U.

Normal: A topological space (5, T) is normal if for every pair of disjoint closed sets A,B $C\overline{S}$ there exists a pair of disjoint open sets U,V € T such that A C U and £ C V.

Open set: If (5,71 is a topological space then all the subsets of T are open.

Second countable: A topological space is second countable if it has a countable basis.

Subspace: A space $T\backslash$ is a subspace of Ti if $T\backslash$ C Ti.

Weaker topology: If (S, TO and (5,7^) arc two topological spaces. $T\backslash$ is said to be weaker than Ti is T_x C T_2.