Open access • Journal Article • DOI:10.1007/S00163-012-0144-Y

Towards an ontology of design: lessons from C–K design theory and Forcing — Source link

Armand Hatchuel, Benoit Weil, Pascal Le Masson

Institutions: Mines ParisTech

Published on: 01 Apr 2013 - Research in Engineering Design (Springer-Verlag)

Topics: C-K theory, Process ontology, Upper ontology, Ontology (information science) and Forcing (recursion theory)

Towards an ontology of design: lessons from C-K design theory and Forcing[1]

Armand Hatchuel[*, 1], Benoit Weil[1], Pascal Le Masson[1]

[*] corresponding author

[1] *Mines ParisTech Centre de Gestion Scientifique*

60 Boulevard Saint Michel, F-75 272 Paris Cedex 06, France

hatchuel@ensmp.fr, bweil@ensmp.fr, lemasson@ensmp.fr

tel: +33 1 40 51 91 09, fax: +33 1 40 51 90 65

Paper submitted for the special issue of RIED on Design theory

Abstract

In this paper we present new propositions about the ontology of design and a clarification of its position in the general context of rationality and knowledge. We derive such ontology from a comparison between formal design theories developed in two different scientific fields: Engineering and Set theory. We first build on the evolution of design theories in engineering, where the quest for domain-independence and "generativity" has led to formal approaches, likewise C-K theory, that are independent of what has to be designed. Then we interpret Forcing, a technique in Set theory developed for the controlled invention of new sets, as a general design theory. Studying similarities and differences between C-K theory and Forcing, we find a series of common notions like "d-ontologies", "generic expansion", "object revision", "preservation of meaning" and "K-reordering". They form altogether an "ontology of design" which is consistent with unique aspects of design.

***Key words**: design theory- innovation- ontology- programming theory*

[1] A paper on C-K theory and Forcing was presented by A. Hatchuel and B. Weil at ICED 07 under the title "Design as Forcing: deepening the foundation of C-K theory". This article extends widely the findings of the former one and has benefited from the discussions that took place with members of the SIG Design theory of the Design society. We acknowledge Yoram Reich and Akin Kazakçi for their useful comments.

Part 1. Introduction

What is design? Or in more technical terms, can we clarify as rigorously as possible some of the main features of an *ontology of design*? In this paper, we develop an approach of such ontology that became possible thanks to the following developments:

- the elaboration in the field of engineering of formal design theories, like C-K theory, (Hatchuel and Weil 2003, 2009) which are *independent* of any engineering domain and avoid too strong restrictions about *what* is designed.

- the exploration of design theories that could have emerged in other fields from a similar process of abstraction and generalization. In this paper, we introduce Forcing (Cohen 1963), a technique and branch of Set theory that generalized *extension procedures* to the creation of new collections of sets. It presents, from our point of view, specific traits of a design theory with highly general propositions.

These design theories offered a unique material for a comparative investigation. The study of their similarities and differences is the core subject of this paper. It will lead us to what can be named *an ontology of expansion* which clarifies the nature of design. This ontology is no more postulated but revealed by common assumptions and structures underlying these design theories. Therefore, our findings only reach the ontological features consistent with existing formalizations of design. Yet, to our knowledge, such ontology of expansion, as well as the interpretation of Forcing as a design theory, had not been investigated and formulated in existing literature.

Before presenting the main hypotheses and the structure of this paper some preliminary material on design theories is worth mentioning.

Formal design theories. In the field of engineering, efforts to elaborate formal (or formalized) design theories have been persistent during the last decades (Yoshikawa 1981; Reich 1995; Braha and Reich 2003). "Formal" means the attempt to reach rigor, if possible logical and mathematical rigor, both in the formulation of hypotheses and the establishment of findings. It also delineates the limited scope and purpose of these theories. Formal design theories (in the following we will say design theories or design theory) are only one part of the literature about design. They neither encompass all the findings of design research (Finger and Dixon 1989; Cross 1993), nor describe all activities involved in design in professional contexts. For instance, it is well known that design practice is shaped by *managerial, social and economic forces* that may not be captured by formal design theories. Yet, this does not mean that design theories have no impact. Such forces are influenced by how design is described and organized. Actually, it is well documented that design theories, in engineering and in other fields, have contributed to change dominant design practices in Industry (Le Masson et al. 2011).

Still, the main purpose of design theory is the advancement of design science by capturing the type of reasoning (or model of thought) which is specific to design. As an academic field, design theory has its specific object and cannot be reduced to decision theory, optimization theory or problem-solving theory. Therefore, recent design theories focus on what is called the creative or "generative" (Hatchuel et al. 2011) aspects of design. Indeed, any design engineer (or designer) uses standard techniques to select or optimise existing solutions. However, design theories target the rationale, the models of thought and reasoning that *only* appear in design. This special attention does not deny the importance of routinized tasks in design, and design theory should, conceptually, account for both creative and routinized aspects of design, even if it does not include all routinized techniques used in design. Likewise, in mathematics, Set theory is able to account for the core assumptions of Arithmetics, Algebra and Analysis, yet it cannot replace these branches of mathematics. Finally, by focusing on creative design, design theory can complement decision theory by helping engineering and social sciences (economics, management, political science…) to better capture the human capacity to intentionally create new things or systems.

Research methodology. For sure, there is no unique way to explore an ontology of design. However, in this paper we explore a research path that takes into account the cumulative advancement of theoretical work in engineering design. The specific material and methodology of this research follows from two assumptions about the potential contribution of design theories to the identification of an ontology of design.

Assumption *1: Provided they reach a high level of abstraction and rigor, design theories model ontological features of design,*

Assumption 2*: Provided there is a common core of propositions between design theories developed in different fields, this core can be seen as an ontology of design.*

An intuitive support for these assumptions and the method they suggest, can be found using an analogy with Physics. If the goal of our research was to find an ontology of "matter" or "time" *consistent with contemporary knowledge* in Physics, a widely accepted method would be to look in detail to common or divergent assumptions about "matter" or "time" in contemporary theories and physics. And clearly, there is a wide literature about the implications of Special relativity and Quantum mechanics for the elaboration of new ontologies of time and matter. Similarly, our method assumes that design theories have already captured a substantial part of our knowledge about design and may be valid guides for the exploration of an ontology of design.

Outline of the paper. In this section (part 1) we outline the trends towards generality and domain-independence followed by design theories in engineering. They are well illustrated by the specific features of C-K theory (Hatchuel and Weil 2003). Then we discuss the existence of design theories in other fields like science and mathematics. We suggest that in Mathematics, Forcing, a method in Set theory developed for the generation of new sets, can be seen as a design theory. In Part 2 and 3 we give an overview of the principles and findings of both C-K theory and Forcing. In part 4, we compare the assumptions and rationale of both theories. In spite of their different contents and contexts, we find that C-K theory and Forcing present common features that unveil an ontology of design that we characterize as an "ontology of expansion".

1.1- Design theories in engineering: recent trends

In engineering, the development of formal design theories can be seen as a quest for more generality, abstraction and rigor. This quest followed a variety of paths and it is out of the scope of this paper to provide a complete account of all theoretical proposals that occurred in the design literature during the last decades. We shall briefly overview design theories which have been substantially discussed in the literature. Still, we will analyse in more detail C-K theory as an example of formal theories developed in engineering that present a high level of generality. Then we will compare it to a theory born in Mathematics.

Brief overview of design theories. In the field of engineering, General Design Theory (Yoshikawa 1981) and Axiomatic Design (AD) (Suh 1990) were among the first to present formalized approaches of design. Both approaches have in common to define the design of a new entity as the search of some *ideal mappings* between required *functions and selected attributes*. The core value of these theories was to model these functions and attributes with mathematical structures that helped to define and warrant specific operations of design. For instance, in GDT, Hausdorff spaces of functions and attributes were assumed. Such mathematical structures offered the possibility to map new desired intersections of functions by the "best" adapted intersection of attributes. In AD, Matrix algebras and information theory were introduced. They are used to model types of mappings between attributes (called design parameters) and functions (called Functional requirements). These matrices define the design output. Thus, ideal designs can be axiomatically defined as particular matrix structures (AD's first axiom) and associated to the ideal information required from the design user (AD's second axiom). GDT and AD were no more dependent on any specific engineering domain but still relied on special mathematical structures that aimed to model and warrant "good" mappings.

After GDT and AD, the discussion about design theory followed several directions. Authors introduced a more dynamic and process-based view of design (eg. FBS, (Gero 1990)); they insisted on the role of recursive logic (Zeng and Cheng 1991) as well as decomposition and combination aspects in design (Zeng and Gu 1999a, b). For this research, we only need to underline that these discussions triggered the quest for more general mathematical assumptions that could: i) capture both mapping and recursive processes within the same framework; ii) account for the "generative" aspect of design. Two recent theories are good examples of these trends.

The first one, called Coupled Design Process (CDP, Braha and Reich 2003), kept the distinction between spaces of functions and spaces of attributes (or design parameters) but it allowed them to evolve dynamically by introducing general topological structures (Closure sets and operators). Thanks to these structures, CDP captured, for instance, the introduction of new functions different from those defined at the beginning of the design process. It also described new forms of interplay between functions and attributes, which could be generated by available data bases and not only by some fixed or inherited definitions of the designed objects. Thus CDP, extended GDT and the idea of a satisfactory mapping was replaced by a co-evolution of functions and attributes. Its mathematical assumptions also accounted for the non-linear and non-deterministic aspects of design.

The second design theory is called C-K theory (Hatchuel and Weil 2003; Hatchuel and Weil 2009). C-K theory is consistent with the dynamics captured by CDP. However it is no more built on the distinction between functions and attributes spaces. Instead, it intends to model the special logic that allows a "new object" to appear. This generative aspect is commonly placed at the heart of the dynamics of design. C-K theory models design as the necessary interplay between known and "unknown" (undecidable) propositions. Attributes and functions are seen as tentative constraints used for the stepwise definition of an unknown and desired object. They also play a triggering role in the production of new knowledge. New attributes and new functions are both causes and consequences of changes in the knowledge available. C-K theory explains how previous definitions of objects are revised and new ones can appear, threatening the consistency of past knowledge. Thus the core idea of C-K theory is to link the definition process of a new object to the activation of new knowledge and conversely.

Authors point out (Hatchuel et al 2011) that all these design theories do not present neither radically different nor contradictory point of views about design. Rather, they can be seen as steps belonging to a same endeavour: develop design theory on more general grounds by seeking *domain independence and increased "generativity":*

- *Domain-independence*. Formal design theories aimed to define design reasoning without any object definitions or assumptions coming from specific engineering domains. Thus design theory evolves towards a discipline that could be axiomatically built and improved through empirical field research or theory-driven experiments. A similar evolution has already happened in the fields of decision science or machine learning.

- *Increased generativity.* Design has often been seen as a sophisticated, ill-structured or messy type of problem solving. This vision was introduced by Herbert Simon, but it needs to be extended by introducing a unique aspect of design: the intention to produce "novel" and creative things (Hatchuel 2002). Authors recently called "*generativity*", this intentional research of novelty and surprises which has driven the development of design theories through more abstract mathematical assumptions (Hatchuel et al. 2011).

This second property of design deserves some additional remarks, because it has crucial theoretical consequences about what can be defined as a design task.

- Intuitively, the formulation of a design task has to maintain *some indeterminacy of its goals and some unknown aspects* about what has to be designed. If there is already a deterministic and complete definition of the desired object, design is done or is reduced to the implementation of a constructive predefined algorithm. For instance, finding the solution of an equation when the existence of the solution and the solving procedure are already known should not be seen as a design task.
- However, in practice the frontiers of design are fuzzy. For instance, one can generate "novel" objects from random variables: is it design? For sure, a simple lottery will not appear as a design process. However, random fractal figures, present to observers complex and *surprising* forms. Yet, surprises are not enough to characterize design. The literature associates design work to the fact that novelty and surprises are: i) *intentionally generated* (Schön 1990); or ii) *if they appear by* accident, they may be used as a resource for the design task (an effect that has been popularised as "serendipity"). Authors already model Design practice as a combination of intentionality and indeterminacy (Gero 1996; Braha and Reich 2003).

Domain independence and the modelling of generative processes are main drivers of recent design theory in engineering. On both aspects, C-K theory can be seen as a good representative of the present stage of abstraction and generality in the field. It will be described in more detail and compared to Forcing for the main purpose of this paper: an investigation of the ontology of design.

1.2. The status of design theory: beyond engineering can we find formal design theories?

As described before, the evolution in engineering has led to design theories that are no more linked to engineering disciplines and domains. This is a crucial observation for our research. If design theory is independent of *what is designed,* an ontology of design becomes possible. Similarly an ontology of decision became possible when decision theory was no more dependent of its context of application. However, the independence from engineering domains does not prove that design theory has reached a level of generality that is acceptable outside engineering. Hence, our research of an ontology of design would be more solidly grounded if we could also rely on design theories that emerged in other fields than engineering. Yet, are there other fields where general design theories can be found? And do they present common aspects with engineering design theories? A complete answer to such questions would require a research program in philosophy, Art and science that is beyond the scope of one paper. Thus, we focused our inquiry on potential design theories in science and mathematics. We will introduce Forcing in Set theory and explain why, from our point of view, it can be seen as a general design theory.

Design theory in science. In standard scientific research, the generation of new models and discoveries is common place. Yet, classically the generative power of science is based upon a dynamic interaction between theory and experimental work. This view of science has been widely discussed and enriched. At the end of the 19th century, the mathematician Henri Poincaré suggested that the formation/construction of hypotheses is a creative process (Poincaré 2007). Since, it has often been argued that the interaction between theories and experiments follows no deterministic path, or that radically different theories could present a good fit with the same empirical observations. Proposed by Imre Lakatos (Worral and Currie 1980), the idea of "research programmes"(which can be interpreted in our view as "designed research programs") seemed to better account for the advancement of science than a neutral adjustment of theory to facts. In Modern Physics (relativity theory or quantum mechanics) the intentional generation of new theories is an explicit process. These theories[2] are expected to be conceived in order to meet specific requirements: consistency with previously established knowledge, unification of partial theories, mathematical tractability, capacity to be tested experimentally, the prediction of new facts, and so forth... Thus, it is acceptable to say that in classic science new theories are designed. However,(Worral and Currie 1980) to our knowledge, there is no formal design theory that has emerged as a general one in this field. This is a provisional observation and further research is needed. However, this state of information contrasts with what can be found in Mathematics where the generation of new objects has been modelled.

Design theory in mathematics: the forcing model in Set theory. Following again Henri Poincaré, it is now widely accepted that mathematical objects are created (i.e. designed) to reach increased generality, tractability and novelty. Yet, these views are not enough to offer a formal design theory. For our specific research program, a chapter of Set theory, called *Forcing*, deserves a special attention because it builds a highly general process for the design of new objects (set models) within the field of Set theory. Forcing generates new sets that verify the axioms of set theory (i.e. new "models" of Set theory). It is also a theory that proves why such technique has general properties and applications. Forcing played a major role in the solution of famous mathematical problems of the 20th century. For instance, we will see how Forcing has been used to generate new real numbers that changed existing ideas about the cardinality - i.e. "size"- of uncountable infinite sets. For sure, Forcing is embedded in the mathematical world of Set theory. However, the level of abstraction of Set theory is such that the following hypothesis can be made and will be justified after a more detailed presentation of Forcing:

Hypothesis: Due to the abstraction of Set theory, Forcing can be seen as a general design theory

1.3. A comparative approach between design theories. Towards a clarification of the ontology of design

If Forcing can be seen as a general design theory, then different design theories could have reached, independently, a high level of abstraction. And if these theories present a *common core of propositions*, this core would be a good description of what design is essentially *about* and what makes design reasoning possible. Thus our research method was not to postulate an ontology of design and discuss its validity but *to infer* it from the comparison of design theories coming from different scientific contexts. In this paper, we focus our comparison on i) C-K theory as a representative of design theories in engineering design; and ii) Forcing as a general design theory in Set theory. This comparison was structured by the following questions:

Q1: What are the similarities and differences between C-K theory and Forcing?
Q2: What are the common propositions between such theories? What does this "common core" tell us about the ontology of design?

In spite of their different backgrounds, we found consistent *correspondences* between both theories. As expected, they offer new ground for the clarification of an ontology of design. However, such comparison presents limitations that have to be acknowledged. Forcing is a mathematical theory well established in the field of Set theory. The scope of C-K theory is broader. It aims to capture the design of artefacts and systems including physical and symbolic components. Its mathematical formalization is still an open research issue. Therefore, in this paper, we only seek for *insights* revealed by the comparison of their *structural assumptions and operations when they are interpreted as* two models of design. Our claim is that a more rigorous discussion about the ontology of design can benefit from such comparative examination of the structure of design theories.

[2] For instance, there is an active quest for new theoretical physics based on String theory that could replace the standard model of particles.

Part 2. C-K theory: modelling design as a dual expansion

C-K theory has been introduced by Hatchuel and Weil (Hatchuel and Weil 2003; Hatchuel and Weil 2009). C-K theory attempts to describe the operations needed to generate *new objects* presenting desired properties. The conversation about C-K theory in the literature treats both its theoretical implications and potential developments (Kazakçi and Tsoukias 2005; Salustri 2005; Reich et al. 2010; Shai et al. 2009; Dym et al. 2005 ; Hendriks and Kazakçi 2010; Sharif Ullah et al. 2011)[3]. In this section we will present the main principles of C-K theory. They are sufficient to study its correspondences with Forcing (more detailed accounts and discussions can be found in (Hatchuel and Weil 2003; Hatchuel and Weil 2009 ; Hendriks and Kazakçi 2010; Kazakçi and Hatchuel 2009)).

2.1. C-K theory: notions and operators.

Intuitive motivation of C-K theory: what is a design task ? C-K theory focuses on a puzzling aspect of design (Hatchuel and Weil 2009): the theoretical and practical difficulties to define *the departure point of a design task*. In professional contexts such departure points are called “specifications”, “programs” or “briefs”. But in all cases, their definition is problematic: they have to indicate some desired properties of an object *without being able to give a constructive definition of this object, or without being able to warrant its existence by pre-existing knowledge*. This explains why a design task cannot be fully captured by the mere task of a mapping between attributes and functions. Design only appears when such mapping is driven by *an equivocal, incomplete, fuzzy or paradoxical formulation*. Thus to better approach design, we need to model a type of reasoning that begins with a proposition that speaks of an object which is desirable, yet partially unknown, and which construction is undecided with available knowledge. But, this intuitive interpretation leads to difficult modelling issues. How can we reason on objects (or collections of objects) which existence is undecidable? Moreover, because the desired objects are partially unknown, their design will require the introduction of new objects or propositions that were unknown at the beginning of the process. The aim of C-K theory was to give a formal account of these intuitive observations and their consequences.

Concept and knowledge spaces. The name “C-K theory” mirrors the assumption that design can be modelled as the interplay between *two interdependent spaces* having different structures and logics: the space of concepts (C) and the space of knowledge (K). “Space” means here collections of propositions that have different logical status and relations. The structures of these two spaces determine the core propositions of C-K theory (Hatchuel and Weil 2003). Space K contains established (true) propositions or propositions with a clear logical status. Space C is the space where the progressive construction of desired objects is attempted. In this space, we find propositions about objects the existence of which is undecided by the propositions available in K: these propositions of C space are called “concepts” in C-K theory. Example of concepts are propositions like “there exists a flying boat” or “there exists a smarter way to learn tennis”. Design begins when one first concept C0 is used as a trigger of a design process. Design is then described as the special transformation of C0 into other concepts until it becomes possible to reject their undecidability by a proof of existence or non-existence in the K-space available at the moment of the proof (the propositions become decidable in the new K-space). The crucial point here is that in space C, the desired unknown objects or (collections of these objects) can only be characterized by comprehensionnally and not extensionally. If a true extensional definition of these objects existed in K or was directly deductible from existing K (i.e. there is a true constructive proof of their existence in K) then the design task would already been done. Now when a new object is designed its existence becomes true in space K, the space of known objects and propositions with a *decided* logical status, ie its concept becomes a proposition of K. To summarize:

- **Space K** contains all established (true) propositions (the available knowledge).

[3] There is also documented material on its practical applications in several industrial contexts (Elmquist and Segrestin 2007; Ben Mahmoud-Jouini et al. 2006; Hatchuel and Weil 2003; Hatchuel et al. 2004, 2006; Gillier et al. 2010; Elmquist and Le Masson 2009).

- **Space C** contains "concepts" which are *undecided* propositions by K (neither true nor false in K) about some desired and *partially unknown* objects x.

If follows from these principles that the structure of C is constrained by the special constructive logic of objects the existence of which is undecided. The structure of K is a free parameter of the theory. This corresponds to the observation that design can use all types of Knowledge. K can be modelled with simple graph structures, rigid taxonomies, flexible "object" structures, special topologies (Braha and Reich 2003) or Hilbert spaces if there are stochastic propositions in K. What counts from the point of view of C-K theory is that the structure of K allows distinguishing between decided and undecidable propositions. Indeed the K space of an engineer is different from an industrial designer's one: the latter may include perceptions, emotions, theories about color and form and this will directly impact the objects they will design but, basically, from the point of view of design theory *their model of reasoning* can be the same.

Concepts, as defined in C-K theory, attempt to capture the ambiguity and equivocality of "briefs" and "specifications". Therefore, concepts are propositions of the form: "*There exists a (non empty) class of objects x, for which a group of properties p_1, p_2, p_k holds in K*"[4]. Because Concepts are assumed as undecidable propositions in K, the collection of objects x that they "capture" has unusual structures. This is a crucial point of C-K theory that can be illustrated with an example E of a design task:

Example E: let us consider the design task E of "new tyres (for ordinary cars) without rubber". The proposition "*there exists a (non empty) class of tyres for ordinary cars without rubber*" is a concept as it can be assumed as undecidable within our present knowledge. For sure, existing tyres for ordinary cars are all made with rubber and there are no existing, or immediately constructible, tyres without rubber. Moreover, we know no established and invariant truth that forbids the existence of such new objects that we call "no-rubber tyres " (Example E will be used as an illustration in all sections of this paper). C-K theory highlights the fact that the design task E (and any design task) creates the necessity to reason *consistently* on "no rubber tyres" which existence is undecidable in K. These objects form *a class* that corresponds to a formula that is undecidable in K.

At this stage, the mathematical formulation of C-K theory is still a research issue and a key aspect of such discussion is the interpretation and formalization of the unknown and undecidable aspects of a "concept"[5]. However, turning undecided concepts into defined and constructible things is required by a design task and it is this process that is tentatively described by C-K theory. Necessarily, these operations are "expansions" in both K and C:
- in K, we can attempt to "*expand*" the available knowledge (intuitively, it means learning and experimenting) if we want to reach a decidable definition of the initial concept
- in C we can attempt to add new properties to the first concept in order to reach decidability. This operation, which we call a partition (see below) is also *an expansion* of the definition of the designed object (see below). If I say that I want to design a boat that can fly, I can logically expect that I have to add some properties to the usual definition of boats.

[4] It can also be formulated as: "The class of objects x, for which a group of properties p1, p2, pk holds in K is non empty".

[5] The literature about C-K theory discusses two ways to treat this issue:

- the class of "non rubber tyres for ordinary cars" can be seen as a special kind of set, called C-set, for which the existence of elements is K-undecidable (Hatchuel and Weil 2009). This is the *core* idea of the theory and the most challenging aspect of its modelling. Clearly assuming "elements" of this C-set will be contradictory with the status of the concept, or we would have to speak of elements without any possibility to define them or to construct them. This is in contradiction with the classic elementarist approach of sets (see Jech, Dehornoy). It means that the propositions "C- set is empty" or "a C-set is non-empty" is K-undecided and only after design is done we will be able to decide this question. Technically, Hatchuel and Weil suggest that C-Sets could be axiomatized within ZF if we reject the axiom of choice and the axiom of regularity, as these axioms assume necessarily the existence of elements. More generally, in space C, where the new object is designed, the membership relation of Set theory has a meaning only when the existence of elements is proved.

- Hendriks and Kazakci (Hendriks and Kazakçi 2011) have studied an alternative formulation of C-K theory only based on first order logic. They make no reference to C-sets and they reach similar findings about the structure of Design reasoning.

The core proposition of C-K theory is that design appears when both expansions interact. And C-K theory studies the special structure and consequences of such interplay.

- The design process: partitions and C-K operators. As a consequence of the assumptions of C-K theory, design can only proceed by a step-by-step "partitioning" of the initial concept or its corresponding class. Due to the undecidability of concepts and associated classes, "partitions" of a concept cannot be complete family of disjoint propositions. In the language of C-K theory, partitions are one or several new classes obtained by adding properties (coming from K) to the existing concepts. If C_k is the concept "*there exists a non empty class of objects which verify the properties p_0, p_1, p_2... and p_k*", a partition will add a new property p_{k+1} to obtain a new concept $C_{k+1.}$ Such partition create a partial order where $C_{k+1} > C_k$. However, in Space C the class associated to C_{k+1} is not included in the Class associated to C_k, as not extensional meaning holds in Space C. There is no warranted existence of any element of a Class associated to a concept. These additions form a "nested" collection of concepts. Beginning with concept C_0, this partitioning operation may be repeated, whenever there is an available partitioning property in K and until the definition of an object is warranted in K.

Having in mind the interplay between C and K, this partitioning process has specific and unique (Ullah et al. 2011) features.
- Each new partition of a concept has an unknown status that has to be "tested" in K. "Testing" means activating new knowledge that may check the status of the new partition (mock-ups, prototypes, experimental plans are usual knowledge expansions related to a group of partitions).
- Testing a partition has two potential outputs: i) the new partition is true or false, thus forms an expansion in K, or is still undecidable and forms an expansion in C; ii) testing may also expand existing knowledge in a way which is not related to the status of the tested partition (surprises, discoveries, serendipity…). Such new knowledge can be used to generate new partitions and so forth... Finally, "testing" the partition of a concept always expands C or expands K by generating new truths. Hence the more we generate unknown objects in C, the more we may increase the expansion of K.

Example E: Assume that the concept of "non-rubber tyres" is partitioned by the type of material that replaces rubber. This depends of the knowledge we have in K about materials: for instance, plastics, metal alloys and ceramics. Thus we have three possible partitions: "non-rubber tyres with plastics", "non-rubber tyres with metal alloys" and "non-rubber tyres with ceramics". These partitions may create new objects. And testing these partitions may lead to new knowledge in K, for instance new types of plastics, or new materials that are neither plastics, metal alloys or ceramics!

By combining all assumptions and operations described in C-K theory, the following propositions hold (Hatchuel and Weil 2003, 2009; Hendriks and Kazakci 2010)
- Space C has necessarily a tree structure that follows the partitions of C_0 (see Fig 1).
- A design *solution* is the concept C_k that is called the first *conjunction* i.e. the first concept to become a true proposition in K. It can also be defined by the series of partitioning properties (p_1, p_2 …p_k) that forms the design *path* that goes from the initial concept C_0 to C_k. When C_k becomes true in K (a design is reached), the class associated to the series of concepts (C_0, C_1, C_2,... C_k) verify the property: $\forall i$, i=0…k-1, $C_k \subset C_{i.}$ [6]ie: it becomes possible to use the inclusion relationship since the existence of the elements of C_k is true in K and these elements are also included in all concepts that are "smaller" than C_k.
- The other classes resulting from partitions of C_0 are concept expansions that do not form a proposition that belongs to K.
- All operations described in C-K theory are obtained through four types of operators within each space and between spaces: C-C, C-K, K-K, and K-C. The combination of these four operators is assumed to capture the specific features of design, including creative processes and seemingly "chaotic" evolutions of a real design work (Hatchuel et al. 2006). From the point of view of C-K theory, standard models of thought and rationality do not model concepts and can be interpreted as K-K operators.

[6] It may be surprising that the inclusion relation becomes possible: it becomes possible only when the existence is proved.

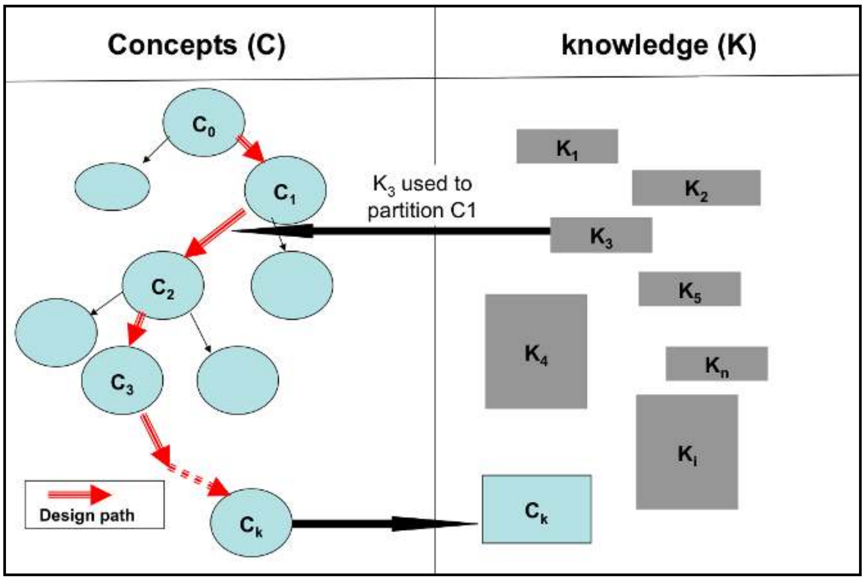

Figure 1: C-K diagram

.

2.2. C-K theory: findings and issues

For sure, neither C-K theory nor any other design theory will warrant the existence of “tyres without rubber”. Design theories, like C-K theory, only model *the reasoning and operations* of the design process and capture some of its “odd” aspects. Thus C-K theory introduces the notion of “expanding partition” which captures a wide range of creative mechanisms.

Expanding partition: generating new objects through chimeras and “crazy” concepts. In C-K theory, it is crucial to distinguish between two types of partitions in space C: *expanding* and *restricting* ones. To do so we need to introduce some additional structures in K: *the definition of known objects*. In example E, the attribute “made with rubber” is assumed to be a common attribute of all known tyres in K. Therefore, the partition “without rubber” is not a known property of the class of objects “tyres”. This partition is called an *expanding partition* as it attempts to *expand* the definition of tyres by creating new tyres, which are different from existing ones. Suppose that the concept is now “a cheaper tyre” and the first partition is “a cheaper tyre using a rubber coloured in white”: if “tyres with white rubber” are known in K, this is called a *restricting* partition. Restricting partitions only act as *selectors* among existing objects in K. While, expanding partitions have two important roles:

- they *revise the definition of objects* and potentially create new ones; they are a vehicle for intentional novelty and surprise in design;
- they *guide the expansion of knowledge* in new directions that cannot be deduced from existing Knowledge.

The generative power captured by C-K theory comes from the combination of these two effects of expanding partitions. Revising the definition of objects allows for new potential objects to emerge (at least as concepts). But this is not enough to warrant their existence in K. Expanding partitions also foster the exploration of new knowledge, which may *help to establish the existence of new objects.* Thus, expanding partitions capture what is usually called *imagination, inspiration, analogies or metaphors*. These are well known ingredients of creativity. However, their impact on design was not easy to assess and seemed rather irrational. C-K theory models these mechanisms as expanding partitions through the old and simple technique of *chimera forming*[7]: partially defining a new object by unexpected attributes (this definition can be seen as crazy or monstruous

[7] The idea of Design as Chimera forming can be traced back to Yoshikawa’s GDT (Yoshikawa 1985) (see the Frodird, p. 177) although the authors didn’t use the term chimera and the theoretical properties of such operations were not fully described in the paper.

regarding existing knowledge). Yet this is only one part of the mechanism. C-K theory unveils two distinct effects of these chimeras: allow for new definition of things and guide the expansion of new knowledge. By disentangling these two roles and the value of their interplay and superposition, C-K theory explains *the rationality of chimeras* and seemingly "crazy concepts" in design: they force the designer to explore new sources of knowledge which could, surprisingly, generate new objects *different* from the "crazy concepts". *It is worth mentioning that this is not classic trial and error reasoning*. Trials are not only selected among a list of predefined possibles. Trials are regenerated through C and K expansions. And the acquired knowledge is not only due to errors but also comes from unexpected explorations. And finally, most of potential trials stay at a stage of chimeras, and yet have generated new knowledge.

Example E: the concept of "non-rubber tyres using plastics" may appear as a chimera and rather "crazy" if known plastics do not fit with usual tyre requirements. But, from another point of view, it may trigger the investigation of plastics offering better resistance. Again, these new plastics may not fit. The same process could happen with ceramics and metal alloys and still only reach undecidable concepts. Meanwhile space K would have been largely expanded: new alloys, new plastics new ceramics and more. Then, and only then, new partitions can appear in Space C, for instance introducing new structures combining multiple layers of different materials and new shapes… The whole logic of space C will change: and the first partitions will no more be on types of materials but on new structural forms that were not known at the beginning of the design process.

An important issue: introducing new objects and the preservation of meaning.
Actually, expanding partitions also raise important issues. If, in example E, design succeeds, then "tyres without rubber" will exist in K. Now, if in K the definition of a tyre was "a special wheel made with rubber", such definition is no more consistent with the new designed object and has to be changed. The design of "tyres without rubber" outdates the old definition of tyres. Yet, revising the definition of tyres may impact other definitions like the definition of wheels, and so on. Thus, the revision of definitions has to be done without inconsistencies between *all* old and new objects in K. Clearly, any design should include *a rigorous reordering of names and definitions in K in order to preserve the meaning of old and new things.* Otherwise definitions will become less consistent and the whole space K will be endangered. Finally, design theory underlines a hidden, yet necessary, impact of design: the perturbation of names and definitions. It warns about the necessity to reorganize knowledge in order to preserve *meaning in K* i.e. the consistency of definitions in K.

What is the generality of the principles and issues raised by C-K theory? Are there implicit assumptions about design that limit the generality of the theory? We now explore existing similarities and differences between C-K theory and a general technique of modern set theory called "forcing". This comparison will guide us towards an ontology of "expansion" as a core ontological feature of design.

Part 3. Design inside Set theory: the Forcing method.

Can we find design theory or methods in mathematics? If a crucial feature of design is the intentional generation of new objects, several design approaches can be found. A branch of mathematics called intuitionism even perceives the mathematician as a "creative subject" (Kazakçi and Hatchuel 2009). Within more traditional mathematics there is a wide variety of "extensions" which can be interpreted as design techniques. "Extension" means transforming some existing mathematical entity M (a group, a ring, a field,…) in order to generate a larger one N, related to but different from M, that contains new entities which verify new selected properties.

The design of complex numbers. Extension procedures are usually dependent of the specific mathematical structure that is extended. Classic maths for engineering includes an example of such ad hoc extensions: the generation of complex numbers from Real ones. The procedure shows clear features of a design process. Real numbers have no "negative squares", yet, we can generate new entities called "complex numbers" that are "designed" to verify such a strange property. The method uses special properties of the division of polynomials. Let us divide any polynomial by a polynomial which has no real root (for instance X^2+1); we generate equivalence classes built on the remainder of the division. The equivalence classes obtained by the polynomial division by X^2+ 1 are all of the form aX+b where (a, b) $\in$ IR^2. These equivalence classes have a field structure (ie with an addition, a multiplication…) and it contains the field of real numbers (all the equivalence classes where a = 0). In this field the polynomial X^2+1 belongs to the equivalence class 0, ie $X^2+1\equiv 0$. Hence the classes can be renamed with the name a*i* +b where *i* verifies $i^2 + 1 = 0$, ie *i* can be considered as the complex (or imaginary) root of $x^2+1=0$. Just like the equivalence classes to which they correspond, these a*i* + b entities form a field (with an addition, a multiplication…), ie a new set of designed numbers which have standard properties of reals plus new ones. It is worth mentioning that with the design of complex numbers the definition of "number", likewise the definition of "tyre" in example E, had to be revised. And the most striking revision is that the new imaginary number *i* is not a real number, yet all the powers of i^2 are reals!

Clearly, this extension method is dependent of the specific algebra of the ground structure ie. the field of real numbers. Therefore, if an extension method acts on general and abstract structures, then it could be interpreted as a general design theory. This is precisely the case of *Forcing* discovered by Paul Cohen in 1963 (Cohen 1963, 1964, 1966)[8]. It generalizes the extension logic to *any sets* and allows the generation of *new collection of sets*. We first present the principles of Forcing to support the idea that Forcing is a design theory; then, we study its correspondence with C-K theory.

3.1. Forcing: designing new collections of sets

Forcing has been described by historians of Set theory as « a *remarkably general and flexible method with strong intuitive underpinnings for extending models of set theory* » (Kanamori 1996). Let us remind what are "models of set theory" before describing the Forcing operations.

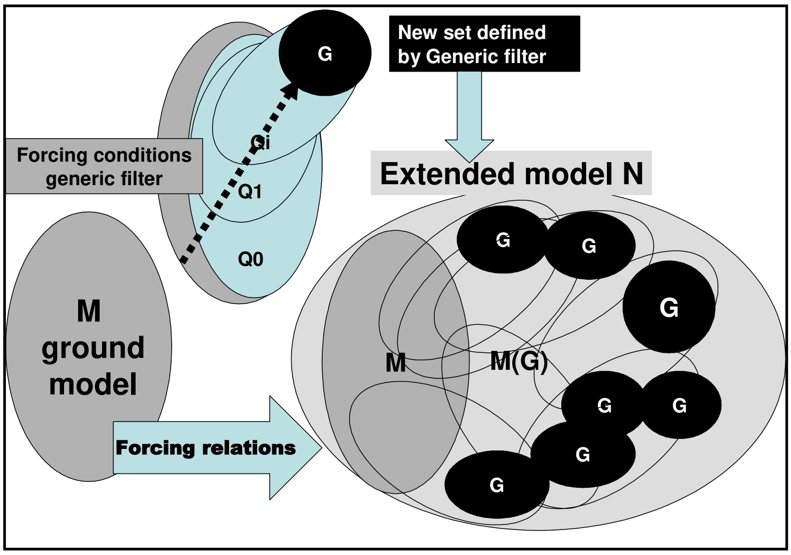
G
New set defined by Generic filter
Forcing conditions generic filter
Qi
Q1
Q0
Extended model N
M ground model
Forcing relations
G
G
G
M
M(G)
G
G
G
G

Figure 2: The forcing Method

- **Models of Set theory.** Set theory is built on a short list of axioms called Zermelo-Frankael axiomatic (ZF) (Jech 2002)[9]. They define the equality, union, separation and well formation of sets. They also postulate the existence of some special sets. A *model of set theory* is any collection of sets that verify ZF; it is also called *a model of ZF*. In the engineering world, the conventional *definition of a thing or a class of things (*for instance, the definition of tyres) plays the role of a "model of tyres" even if real life conventions are less demanding than mathematical ones; thus, a

[8] He has been awarded a Fields medal for this work.

[9] Properly speaking, ZF has infinitely many axioms: its axiomatization consists of six axioms and two axiom schema's (comprehension and replacement), which are infinite collections of axioms of similar form. We thank an anonymous reviewer for this remark.

model of tyres is *a collection of sets of tyres* that verify the usual definition of tyres[10]. In the industrial world, thanks to technical standards, most engineering objects are defined through models (for example, machine elements).

- Why Forcing? Independent and undecidable propositions in Set theory. After the elaboration of ZF, set theorists faced propositions ($\mathcal{P}$), like the "axiom of choice" and the "continuum hypothesis"[11], that seemed difficult to prove or reject within ZF. This difficulty could mean that these propositions were *independent* from the axioms of ZF, hence were *undecidable* within ZF, so that models of ZF could *verify or not* these propositions. Now, proving the existence of a model of ZF that does not verify the axiom of choice is the same type of issue than proving that there is a model of tyres with no rubber. One possible proof in both cases is *to design such a model*! Actually, designing new models of ZF is not straightforward and there comes Forcing, the general method invented by Paul Cohen.

3.2. The forcing method: ground models, generic filters and extensions

Forcing assumes the existence of a first model M of ZF, called *the ground model* and then it offers *a constructive procedure* of a new model N, called the extension model, different from M, which refutes or verifies $\mathcal{P}$ and yet, is a model of ZF. In other words, Forcing generates new collections of sets (i.e. models) and *preserves ZF.* Hence, it creates new sets but preserves what can be interpreted as their *meaning* i.e. the basic rules of sets. Forcing is not part of the basic knowledge for engineering science and is only taught in advanced Set theory courses. Therefore, a complete presentation of Forcing is beyond the scope of this paper and we will avoid unnecessary mathematical details and focus on the most insightful aspects of Forcing[12] needed to establish the findings of this paper. Moreover, it is precisely because Forcing is a very general technique that one can understand its five main elements and its logic without a complete background in advanced Set Theory.

- *The first element* of Forcing is ***a ground model M***: a well formed collection of sets, a model of ZF.
- *The second element* is the set of ***forcing conditions*** that will act on M. To build new sets from M, we have to extract elements according to some *conditions* that can be defined in M. Let us call (Q, <) a set of candidate conditions Q and *a partial order relation* < on Q. This partially ordered set (Q, <) is completely defined in M. From Q, we can extract conditions that can form series of *compatible and increasingly refined conditions* (q_0, q_1, q_2 ... q_i) with for any i: $q_i < q_{i-1}$; this means that each condition *refines* its preceding one. The result of each condition is a subset of M. Hence the series (q_i) builds series of nested sets, each one being included in its preceding set of the series. Such series of conditions generates a *filter*[13] F on Q. And a filter can be interpreted as *a step-by-step definition of some object or some set of objects* where each step refines the preceding definition by adding new conditions.

- *The third element of Forcing* is the **dense subsets** of (Q, <): a *dense* subset D of Q is a set of conditions so that any condition in Q can be refined by at least one condition belonging to this dense subset. *One property of dense subsets* is that they contain very long (almost "*complete") definitions of things (or sets) on M, since every condition in Q, whatever its "length",* can always be refined by a condition in D.

[10] Such models of things are also present in Design theories (see for instance the "entity set" in GDT (Yoshikawa 1981)).

[11] The two propositions of this type that gave birth to the forcing method are well known in set theory. The first one is "every set of nonempty sets has a choice function"; the second one is the existence of infinite cardinals that are intermediate between the cardinal of the integers and the cardinal of the reals also called the continuum hypothesis.

[12] Complete presentations of Forcing can be easily found in standard textbooks in advanced set theory (Kunen 1980; Jech 2002; Cohen 1966)

[13] Filters are standard structures in Set theory. A filter F is a set of conditions of Q with the following properties: non empty; nestedness (if p < q and p in F then q is in F) and compatibility (if p, q are in F, then there is s in F such that s < p and s < q).

- *The fourth element* of Forcing (its core idea!) is the formation of a ***generic filter*** **G** which *step by step* completely defines **a new set not in M !** Now how is it possible to jump out the box M? Forcing uses a very general technique: it *creates an object that has a property that no other object of M can have !* (Remark: this is similar to an expanding partition in the language of C-K theory). Technically, a generic filter is defined as a filter that *intersects all dense subsets*. In general this generic filter defines a new set that is not in M[14] but is still defined by conditions from Q, defined on M. Thus, G builds a new object that is necessarily different from all objects defined in M. We can interpret G as a collector of all information available in M in order to create something new not in M.

- The *fifth element* of Forcing is the construction method of the **extended model N**. The new set G is used as *the foundation stone* for the generation of new sets combining systematically G with other sets of M (usually called M(G)). The union of M and M(G) is the extension model N. (Fig 2 illustrates how G is built with elements of M, yet G is not in M; then N is built with combinations of G and M). A crucial aspect of Forcing is the necessity to well organize the *naming* of the sets of M when they are embedded in the extension set N. Thus, elements of M have two names, the old one and the new one. The generic set G, the new designed object, has one unique name as it was not present in M.

The main Forcing theorems. Paul Cohen invented Forcing and proved a series of theorems that highlighted the generality of the design process. The main results can be synthesized as follows:

- Forcing preserves ZF: Whenever a generic filter G exists, the new model N is a model of ZF. Hence, ZF is preserved and the new sets are not meaningless objects.

- Forcing controls all properties of N: All properties of the elements of N are strictly dependent on the conditions (q_0... q_i) that formed the generic filter. This means that any true proposition T in N is such that there exists some p_i in G so that: $q_i \rightarrow$ T. Hence, the appropriate generic filter G warrants the existence of new models of sets with desired properties.

The impact of Forcing on Set theory has been paramount, and at the same time, historians of mathematics acknowledge its surprising power. « *Set theory had undergone a sea-change and beyond how the subject was enriched it is difficult to convey the strangeness of it"* (Kanamori 1996).

[14] G is not in M as soon as Q follows the splitting condition: for every condition p, there are two conditions q and q' that refine p but are incompatible (there is no constraint that refine q and q'). Demonstration (see (Jech 2002), exercise 14.6, p. 223): Suppose that G is in M and consider D = Q \ G. For any p in Q, the splitting condition implies that there q and q' that refine p and are incompatible; so one of the two is not in G hence is in D. Hence any condition of Q is refined by an element of D. Hence D is dense. So G is not generic.

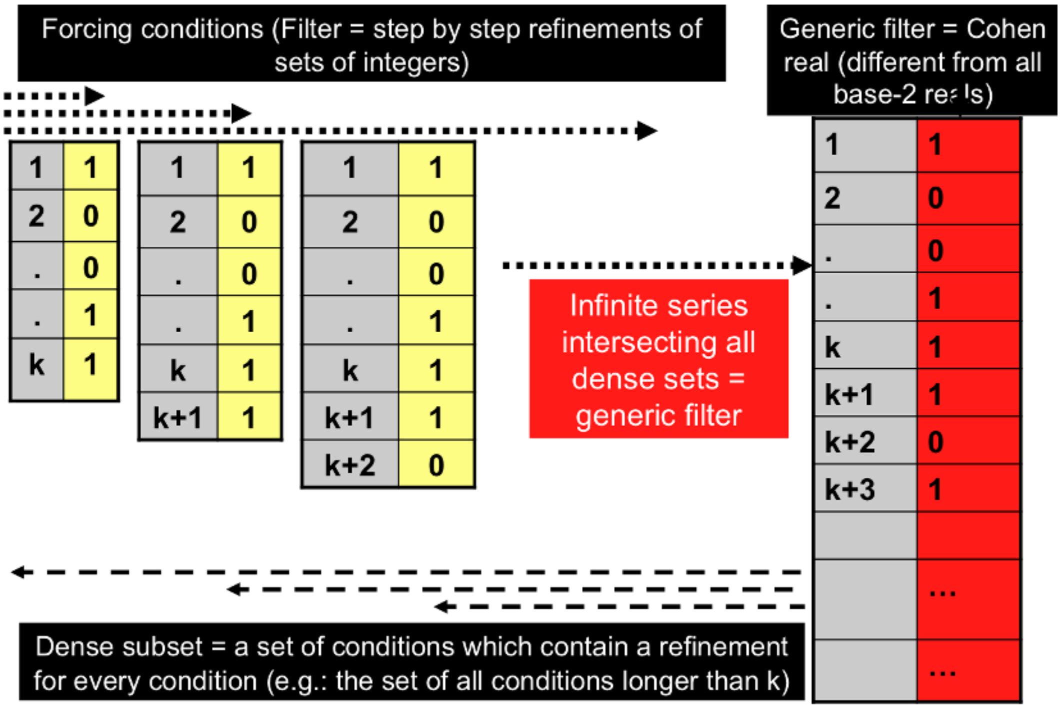
Forcing conditions (Filter = step by step refinements of sets of integers)
Generic filter = Cohen real (different from all base-2 reals)
1 1
2 0
. 0
. 1
k 1
k+1 1
k+2 0
k+3 1
...
...
Infinite series intersecting all dense sets = generic filter
Dense subset = a set of conditions which contain a refinement for every condition (e.g.: the set of all conditions longer than k)

Figure 3: The generation of Cohen Reals by Forcing

3.3. An example of Forcing: the generation of new real numbers.

To illustrate Forcing we give a simple application due to Cohen (Jech 2002). It is the forcing of real *numbers* from integers (see Fig 3).

Ground model: The sets of integers (power set based on the set of integers ω);

Forcing conditions Q: the conditions *Q* can be written as a (0; 1)-functions defined on a subset of ω: assume, a finite series of ordered integers (1, 2, 3, 4,…, k); to each integer assign a (0,1) value; we obtain a new k-list (0, 1, 1, 1,…, 0). The condition is defined over the k first integers and among these integers it extracts some integers (those with value 1) and leaves the others (value 0). It also describes the set of all numbers beginning by the sequence of selected integers. This can be assimilated to the reals written in base 2 and beginning with the same k first binary digits. Then, let us build a more refined condition by keeping this first list and assigning to k+1 a (0,1) value, *without changing the values of the preceding n-list*. We obtain a new condition of length k+1 that refines the first one. The operation can be repeated infinitely. This extension defines the order relation on the conditions *Q*. Note that (*Q*, <) follows *the splitting condition*: for any condition p, (q(0), q(1),… q(k)), there are always two conditions that refine p and are incompatible: (q(0), q(1),… q(k), 0) and (q(0), p(1),… q(k), 1). A series of ordered conditions from length 1 to length k forms a filter; all sets of conditions that contain a refinement of every condition are *dense subsets*;

Generic filter: it is formed with the infinite series of conditions that intersects all dense subsets. Hence, the generic filter G builds an infinite list of selected integers and G is not in M. This follows directly from the splitting condition (see footnote 14) or this can also be demonstrated as follows: for any 0-1 function g in M, Dg = {q ∈ Q, q ⊄ g} is dense so it meets G so that G is different from any g. Hence G forms a new real number (this is the demonstration given in (Jech 2002)). Note that any real number written in base 2 corresponds to a function g. *Hence it means that G forms a real number that is different from any real number written in base 2*[15, 16].

[15] To give a hint on this strange property and its demonstration: Cohen follows, as he explains himself, the reasoning of Cantor diagonalization. He shows that the "new" real is different from any real g written in base 2 by showing that there is at least one condition in G that differentiates G and this real (this corresponds to the fact that G intersects Dg, the set of conditions that are not included in g).

[16] Forcing is a mathematical tool that can Design new sets using infinite series of conditions. In real Design, series of conditions are not always infinite.

Part 4: C-K theory and Forcing: a correspondence that uncovers an ontology of expansion

Now that we have presented C-K theory and Forcing we can come back to our hypotheses and research questions.

4.1- Forcing as a general design theory

The previous brief introduction to Forcing brings enough material to discuss our claim that Forcing is a general design theory, not an ad hoc technique.

- *Design task*: like any design project Forcing needs targeted properties for the new sets to be generated. However, Forcing gives no recipe to find generic filters for all desired properties about sets. It only explains *how such generation is conceivable* without creating nonsense in the world of Sets.
- *Generality*: Forcing uses only universal techniques like *the definition of a new "thing" through a series of special refinements*. Indeed, the basic assumptions of Forcing are the axioms of Set theory and the existence of ground models of sets. However, Set theory is one of the most universal languages that are available.
- *Generativity*: Novelty is obtained by a general method called *generic filter which is independent of the targeted sets*. The generic filter builds a new set that is different from any set that would have been built *by a classic combination of existing conditions within M*. Thanks to this procedure, the generic filter is different from any such combination. Thus genericity creates new things by *stepping out the combinatorial process within M*.

These three observations support the idea that Forcing can be interpreted as a general design theory. Indeed the word "Design" is not part of the Forcing language and it is the notion of "extension" that is used in Forcing and other branches of mathematics. But it is precisely the aim of a design science to unify distinct procedures that appear in different fields under different names, if they present an equivalent structure. Such unification is easier when we can compare abstract and general procedures. And Forcing shows that, likewise design theories in engineering, extensions in mathematics have evolved towards more general models[17]. Let us now come back to our comparison and establish similarities and differences between both theories and why these reveal specific ontological elements of design.

4.2- C-K theory and Forcing: similarities and differences

At first glance, both theories present a protocol that generates new things that were not part of existing background. Yet, similarities and differences between these approaches will lead us to highlight common features which may be explicit in both approaches; or implicit in one and explicit in the other. As main common aspects we find: *Knowledge expandability, Knowledge "voids", and generic expansions*. They form altogether a basic substrate that makes design possible and unique.

a) Knowledge expandability, invariant and designed ontologies (d-ontologies)

Knowledge expandability. Clearly, the generation of new objects needs new knowledge. In C-K theory it is an explicit operation. C-K theory assumes knowledge expansions that are not only the result of induction rules, which can be interpreted as K→K operations. New knowledge is also obtained by C→K and K→C operators which have *a triggering and guiding role through the formation of expanding partitions and concepts* (Kazakçi and Hatchuel 2009). But where is *such new knowledge in Forcing*? Is this a major difference between the two theories? As already remarked by Poincaré (Poincaré 2007), one essential method to create novelty in mathematics is the introduction of *induction rules* which generate actual infinites. In Set theory, the axiom of

[17] There are several forms of extensions in Mathematics that cannot be even mentioned in this paper. Our claim is that Forcing, to our knowledge, presents the highest generality in its assumptions and scope.

infinity allows such repeated induction and plays the role of *an endless supplier of new objects*. Without such expansion technique, generic filters are impossible and Forcing disappears. Thus, both theories assume a mechanism for K expandability even if they use different techniques to obtain it.

Invariant ontologies and the limits of design. The background of Forcing is Set theory. Actually, Forcing creates new models of ZF, *but ZF is explicitly unchanged by Forcing*. The existence of such invariant structures is implicit in C-K theory and relates to implicit assumptions about the structure of K. C-K theory lacks *some explicit rules* about knowledge: at least, some minimal logic and basic wording that allows consistent deduction and learning. These common rules are thus necessary to the existence of design and, like ZF, may not be changed by design. Yet, it is really difficult to establish ex ante what are the invariant rules that *should never be changed by design*. This issue unveils an interesting *ontological limitation* for a design theory. To formulate a design theory we need a minimal language and some pre-established knowledge that is invariant by design ! Intuitively, we could expect that the more this *invariant ontology is general*, the most generative will be the design theory. But it could be argued that too minimal invariant ontology would hamper the creative power of design. We can only signal this issue that deserves further research.
By constrast, we can also define *variable ontologies, i.e.* all definitions, objects and rules that can be changed by design. These variable ontologies correspond to the classic definition of *ontologies* in computer science or artificial intelligence (Gruber 2009). They are generated and renewed by design: we suggest to call them *designed ontologies or d-ontologies,* to remind that they result from a previous design process. Most human Knowledge is built on such d-ontologies. Finally, the ontology of design highlights a specific aspect of Knowledge. It is not only a matter of truth or validity. Design science also discusses how knowledge shapes or is shaped by the frontier between what is invariant or designed at the ontological level.

Example E: In the Tyre industry, rubber should be seen as a designed ontological element of Tyres. Yet, for obvious economic and managerial reasons, it could be considered as an invariant one. In any human activity, the frontier between what is invariant and what can be changed by design is a tough and conflictual issue. The role of design theory is not to tell where such frontier should be, but to establish that the existence of such frontier is an invariant ontological element of design, in all domains, be it in Mathematics, or in engineering.

Beyond invariant ontologies, design needs to generate designed ones and this needs another interesting aspect of knowledge: the presence of knowledge “voids”.

b) Knowledge voids: undecidability and independence

In Set theory, Forcing is used to design models of Sets such that some models satisfy the property $\mathcal{P}$ while others verify its negation. These models prove that $\mathcal{P}$ is *undecidable* within Set theory. When this happens, $\mathcal{P}$ can be interpreted as a *void* in the knowledge about Sets. Conversely, the presence of such voids is a condition of Forcing. In C-K theory, concepts are also undecidable propositions that can be similarly seen as voids. Yet, their undecidability is assumed and they are necessary to start and guide the design process[18]. Thus knowledge voids are a common ontology in both theories. Their existence, detection and formulation is a crucial part of the ontology of design. The word “void” is used as a metaphor, that conveys the image that these “voids” have to *be intentionally “filled” by design*[19]. As proved in Forcing, they signal the existence of *independent structures* in existing knowledge or in a system of axioms.

Example E. If one succeeds to design “Tyres without rubber”, it will be confirmed that: i) the concept of “tyres without rubber” was undecidable within previous knowledge; ii) that the d-ontology of tyres has become *independent* from the d-ontology of rubber.

[18] In a simulation study of C-K reasoning (Kazakçi et al. 2010) voids could be modelled, since knowledge was assumed to have a graph structure

[19] One can also use the image of a “hole”. The metaphor of “holes” has been suggested by Udo Lindemann during a presentation about “Creativity in engineering”. (SIG Design theory Workshop February 2011). It is a good image of the undecidable propositions, or concepts in C-K theory, that trigger a Design process. Udo Lindemann showed that such “holes” can be detected with engineering methods when they are used to find Design ways that were not yet explored.

Thus C-K theory and forcing present consistent views about undecidability and highlight its importance for a science of design: *it is both a necessary hypothesis for starting design (C-K theory) and an hypothesis that can only be proved by design (Forcing).*

This finding leads to three propositions that explain why an ontology of design was so specific and required modelling efforts:

- The ontology of design is not linked to *the accumulation of knowledge*, but to the formation of *independent structures* (voids) in cumulated knowledge.
- The specific rationality of design is to "fill" such voids in order to create desired things – "filling" means to prove the independence between two propositions in K.
- The existence of such desired things remains undecidable as long as they are not designed.

c) Design needs generic processes for expansion

Generic and expanding expansions. In C-K theory, a design solution is a special path (C_0,… C_k) of the expanded tree of concepts in space C. This design path is obtained through a series of refinements which *form a new true proposition in K*. Whenever this series is established, several results hold. The partitions that form the design solution are proved *compatible* in K and define a new class of objects which verifies the first C_0 (initially undecidable in K). Comparing with Forcing, this design path is *also a filter* as the path is generated by a step by step refinement process. It is also a *generic filter in C*[20]. Hence, the design path is a generic filter in C, which includes C_0 and "forces" a new set of objects that verify C_0. Yet the generation of novelty in C-K is not obtained *by the mathematical chimera of an actual infinity of conditions* like in Forcing. It is warranted by: i) the assumption of C_0 as an undecidable proposition in K at the beginning of design; and ii) at least one *expanding partition* and one expansion in K, which are necessary to form one new complete design path. Thus, genericity also exists in C-K theory but it is not built by an infinite induction but by *introducing new truths* and *revising* the definition of objects. Finally, C-K theory and Forcing differ by the technique that generates novelty but both can be seen as *generic expansions* as they are obtained by expansions designed to generate an object that is different from any existing object or any combination of existing objects. Thus, *generic expansions are a core element of the ontology of design.*

Expanding partitions as potential forcings: C-K theory adopts a "real" world perspective. All Knowledge is not given at the beginning of the design process and C-K operators aim to expand this knowledge. This can be interpreted, *yet only as a metaphor*, with the forcing language. We could say that *expanding partitions* offer new *potential forcing conditions.* However, increasing potential Forcings is possible only if expanding partitions are not rejected in K because they contradict some invariant ontology.

K-reordering, new namings and preservation of meaning. In both Forcing and C-K theory, design generates new things. In mathematics, the generation of new? real numbers by Forcing obliged to rediscuss the cardinality of the continuous line. We call K-reordering, these K→K operations that are needed to account for the safe introduction of new objects with all its consequences. For instance, design needs new *names* to avoid confusion and distinguish new objects. Interpretation rules will be necessary to preserve meaning with old and new names. As mentioned before, such issues are *explicitly addressed in Forcing*[21]. In the first formulations of C-K theory (Hatchuel and Weil 2003; Hatchuel and Weil 2009), K-reordering was implicit. Now it is clear that it should receive explicit attention in any design theory. To avoid creating nonsense, design needs such careful K-reordering. These theoretical findings have been confirmed by empirical observations of design teams. During such experiments, authors observed the generation of "noun phrases" (Mabogunje and Leifer 1997): this is the response of designers facing the need

[20] Proof: for any dense subset D of C, there is a refinement of C_k that is in D. But since C_k is also in K, any refinement of C_k is in K and cannot be in C. Hence C_k is in D.

[21] The output of Forcing is not one unique new Set G, but a whole extended model N of ZF. The building of the extension Model combines subsets of the old ground Model M and the new set G. Thus new names have to be carefully redistributed so that an element M with name a gets a new name a' when considered as an element of the new set. As a consequence of these preserving rules the extension Set is well formed and obeys ZF axioms (Jech 2000).

to invent new names to describe objects defined by unexpected series of attributes. These "noun phrases" also allow some partial K-reordering that preserves meaning during the conversations at work.

Part 5: Discussion and conclusion: an ontology of design.

In the preceding sections we have compared two design theories coming from different fields. Our main assumptions were that these theories were sufficiently general to bring solid insights about what is design and what are some of its ontological features. We also expected that these common features would appear when each design theory is used to mirror the other.

What we have found is that an ontology of design is grounded on *an ontology of expansion*. This means that in any design domain, model or methodology we have to find a common group of basic assumptions and features that warrant a consistent model of expansion. Or, to put it more precisely: if we find a reasoning process where these features are present, we can consider it as a design process. What are these features? We have assumed that these features can be inductively obtained from the comparison between two general design theories in different fields. We found six ontological features that we summarize in the first column of table1, where we recall the corresponding elements of each feature for both Forcing (column 2) and C-K theory (column 3).

Ontology of design	Forcing	C-K theory
Invariant ontologies	Axioms of Set theory	Basic logic and language; invariant objects (frontier)
designed ontologies	New models of Sets	New families of objects
Knowledge expansions	Inductive rules (axiom of infinity)	Discovery or guided exploration
"voids", undecidability and independence	Independant axioms Set theory	Concepts and independent structures in K
Generic expansions (generating new thing)	Generic filter	Design path with expanding partitions and K-expansions
K-reordering, naming and preservation of meaning	Building rules for the extension model	New names and reorganising the definition of designed ontologies

Table 1: Ontology of design as a common core of design Theories

These findings have several implications and open areas for further research that we briefly discuss now.

- An ontology of design needs a dynamic frontier between invariant ontologies and designed ontologies. This proposition has important implications for the status of design. *Design cannot be defined as an applied science or as the simple use of prior knowledge*. Invariant ontologies can be seen as some sort of universal lawts. Yet designed ontologies are not deduced from these laws, their design needs extra knowledge and revised definitions. Moreover, it is not possible to stabilize ex ante the frontier between these two ontologies. Fore sure generic expansions need some minimal and invariant knowledge. But design theories say nothing about what could be such minimal frontier. Take the field of contemporary Art, even if Art work was not studied in this research, we can conjecture that invariant ontologies that bear on present artistic work are rather limited . Each artist can design and decide what should stay as an invariant ontology for her own future work. In mathematics we can find similar discussions when axiomatics and foundations are in debate. Therefore, an ontology of design may contribute to the debate about the creative aspects of mathematical work (Kazakçi and Hatchuel 2009). Applying such categories to analyse our own work, we have to acknowledge that the ontology of expansion that we have found is a designed and not an invariant one. It depends on the design theories that we have compared in this paper. New ontological features of design may appear if we study other theories. However, by grounding our work on theories that present a high level of generality, but we can reasonably expect that we have at least captured some invariant features of design.

- ***An ontology of design acknowledges for voids in knowledge: modelling unknowness.*** The notion of "voids" opens a specific perspective on knowledge structures. It should not be confused with the usual "lack of knowledge" about something that already exists or is well defined. It is correct to say that: "we lack knowledge about the presence of water on Mars". In this sentence the notions of presence, Mars, and water have not to be designed. Instead, knowledge voids designate unknown entities which existence requires design work. Thus it is not consistent, from our point of view, to say that "we lack knowledge about tyres without rubber". If we want to know something about them, we have to design them before! These findings open difficult questions that need further research: can we detect all "voids" in Knowledge? Are there limits to such inquiry? Are there different possibilities to conceptualize this metaphor? In our research we modelled "voids" with notions like undecidability and independence which are linked to the common background of C-K theory and Forcing. To challenge these interpretations further is needed to explore new models of what we called "concepts" and "unknown objects". A similar evolution happened with the notion of uncertainty, which was traditionally modelled with probability theory before more general models where suggested (like possibility theories).

- An ontology of design needs generic processes for the formation of new things. An important finding of our comparison is that generating new things needs g*eneric expansions,* which are neither pure imagination nor pure combination of what is already known. What we have found is that design needs a specific s*uperposition and interplay* of both chimeras and knowledge expansions. C-K theory insists on the dual role of expanding partitions which allows to revise the identity and definition of objects. Forcing is not obtained by a finite combination of elements of the ground model. It needs first to break the ground model by building a new object, the generic filter, and then to recombine it with old ones. This was certainly the most difficult mechanism to capture. Design theories like C-K theory and Forcing clarify such mechanisms but they are difficult to express with ordinary language. Our research shows that common notions like idea generation, "problem finding" or "serendipity" are only images or elements of a more complex cognitive mechanism. Indeed, it is the goal of theories to clarify what is confused in ordinary language and design theories have attempted to explain what remained usually obscure in Design. Still, it is a challenge to account more intuitively for the notion of generic expansion.

-An ontology of design needs mechanisms for preservation of meaning and Knowledge re-ordering. This finding signals the *price to pay* if we want to design and to continue expanding knowledge and things. At the core of these operations we find the simplest and yet most complex task: consistent *naming.* It is naming that controls the good reordering of knowledge when design is active. Naming is also necessary to accurately identify new "voids" i.e. new undecidable concepts or independent knowledge structures. Naming is also a central task for any industrial activity and organisation. An ontology of expansion tells us that the most consistent way to organize names is to remember how the things we name have been designed and thus differentiated from existing things. Yet, in practice names tends to have an existence of their own and it is well documented that this contributes to fixation effects (Jansson and Smith 1991). It is also documented that in innovative industries, engineering departments are permanently producing a flow of new objects, thus a complete K-reordering becomes almost impossible and this process continuously *threatens* the validity of naming and component interchangeability (Giacomoni and Sardas 2010).

Limitations and further research. To conclude we must stress again that our findings are limited by our material and research methodology. Our comparative work could be extended and strengthened by introducing other formal design theories, provided they are more general than C-K theory and Forcing and reveal new ontological features.

An alternative to our work would be to study design from the point of view of *its reception* which can be interpreted as a continuation of design or as a K-reordering process, both taking place beyond the designer's work (by clients, users, experts, critics, media, etc..). There is also a wide scientific work on perception that has influenced many designers like for instance, Gestalt theory, contrast and color theory etc… One issue for further research could be to compare the ontology of expansion that we have found for Design to existing ontologies *of perception.*

We also acknowledge that, for instance, social or psychological approaches of design could lead to different perspectives on what design is about. However, the clarification of an *ontology* of design may contribute to new *explorations of the social and psychological conditions of design*. The frontier between invariant and designed ontologies can be interpreted from a social perspective. Design, as we have find it, requires *consistent naming and K-reordering* and this also means special social work and training are needed for the acceptance of design activities. Human Societies need both invariance and evolution. Words, rules, habits, cannot change too rapidly but they also need to evolve by design. Thus one can ask if there are social systems that are more or less consistent with the ontology of expansion that we have described. Social and psychological structures indeed play an important role in *the fixation* of ontologies and in design training and learning. It will be the task of future research to link such theoretical advances to more empirical observations of design tasks (Agogué et al. 2011).

Implication for design practice. The practical lesson of this theoretical research is rather simple. According to our findings design has a specific ontology, anchored in subtle and difficult cognitive mechanisms like Knowledge voids, generic expansions and K-reorderings. Thus we can better understand why design practice can be disconcerting, controversial and stressful; also why empirical design research is so demanding and complex (Blessing 2003). The good news is that design theory can cope with the cognitive “chaos” that seems to emerge from design. We understand that design corresponds to a type of rationality that cannot be reduced to standard learning or problem solving. The rationality of design is richer and more general than other rationalities. It keeps the logic of intention but accepts the undecidability of its target; it aims exploring the unknown and it is adapted to the exploitation of the emergent. Yet, its ontology can be explained, and as any other science, design science can make the obscure and the complex clearer and simpler.

References.

Agogué M, Cassotti M, Kazakçi A (2011) The Impact of Examples on Creative Design: Explaining Fixation and Stimulation Effects. Paper presented at the International Conference on Engineering Design, ICED'11, Technical University of Denmark,

Ben Mahmoud-Jouini S, Charue-Duboc F, Fourcade F (2006) Managing Creativity Process in Innovation Driven Competition. In: Verganti R, Buganza T (eds) 13th International Product Development Management Conference, Milan, 2006. EIASM & Politecnico di Milano, pp 111-126

Blessing LT (2003) What is Engineering Design Research? In: International Conference on Engineering Design, Stockholm, Sweden, 2003.

Braha D, Reich Y (2003) Topologial structures for modelling engineering design processes. Research in Engineering Design 14 (4):185-199.

Cohen PJ (1963) The independence of the Continuum Hypothesis. Proceedings of the National Academy of Science 50:1143-1148.

Cohen PJ (1964) The independence of the Continuum Hypothesis II. Proceedings of the National Academy of Science 51:105-110.

Cohen PJ (1966) Set Theory and the Continuum Hypothesis. Addison-Wesley,

Cross N (1993) Science and design methodology: A review. Research in Engineering Design 5 (2):63-69.

Dym CL, Agogino AM, Eris O, Frey D, Leifer LJ (2005) Engineering Design Thinking, Teaching, and Learning. Journal of Engineering Education January 2005:103-120.

Elmquist M, Le Masson P (2009) The value of a ‘failed’ R&D project: an emerging evaluation framework for building innovative capabilities. R&D Management 39 (2):136-152.

Elmquist M, Segrestin B (2007) Towards a new logic for Front End Management: from drug discovery to drug design in pharmaceutical R&D. Journal of Creativity and Innovation Management 16 (2):106-120.

Finger S, Dixon JR (1989) A Review of Research in Mechanical Engineering Design. Research in Engineering Design 1:51-67 (part I) and 121-137 (Part II).

Gero JS (1990) Design prototypes: a knowledge representation schema for design. AI Magazine 11 (4):26-36.

Gero JS (1996) Creativity, emergence and evolution in design: concepts and framework. Knowledge-Based Systems 9 (7):435-448.

Giacomoni G, Sardas J-C (2010) P.L.M et gestion des évolutions de données techniques : impacts multiples et interchangeabilité restreinte. In: Systèmes d'Information et Management, 2010.

Gillier T, Piat G, Roussel B, Truchot P (2010) Managing innovation fields in a cross-industry exploratory partnership with C-K design theory. Journal of product innovation management Accepted - To be published.

Gruber T (2009) Ontology. In: Liu L, Özsu T (eds) Encyclopedia of Database Systems. Springer Verlag,

Hatchuel A (2002) Towards Design Theory and expandable rationality: the unfinished program of Herbert Simon. Journal of Management and Governance 5 (3-4):260-273.

Hatchuel A, Le Masson P, Reich Y, Weil B (2011) A systematic approach to design theories using generativeness and robustness. In: International Conference on Engineering Design, ICED11, Technical University of Denmark, 2011. p 12

Hatchuel A, Le Masson P, Weil B (2004) C-K Theory in Practice: Lessons from Industrial Applications. In: Marjanovic D (ed) 8th International Design Conference, Dubrovnik, 18th-21st May 2004, 2004. pp 245-257

Hatchuel A, Le Masson P, Weil B (2006) The design of science based-products: an interpretation and modelling with C-K theory. In: Marjanovic D (ed) 9th International Design Conference, Dubrovnik, 15th-18th May 2004, 2006. pp 33-44

Hatchuel A, Weil B (2003) A new approach to innovative design: an introduction to C-K theory. In: ICED'03, August 2003, Stockholm, Sweden, 2003. p 14

Hatchuel A, Weil B (2009) C-K design theory: an advanced formulation. Research in Engineering Design 19 (4):181-192.

Hendriks L, Kazakçi AO (2010) A formal account of the dual extension of knowledge and concept in C-K design theory. Paper presented at the International design conference - Design 2010, Dubrovnik, Croatia,

Hendriks L, Kazakçi AO (2011) Design as Imagining Future Knowledge, a Formal Account. In: Grossi D, Minica S, Rodenhäuser B, Smets S (eds) Logic and Interactive Rationality. pp 111-125

Jansson DG, Smith SM (1991) Design Fixation. Design Studies 12 (1):3-11.

Jech T (2002) Set Theory. Springer Monographs in Mathematics, 3rd millenium edition, revised and expanded edn. Springer, Berlin

Kanamori A (1996) The Mathematical Development of Set Theory from Cantor to Cohen. Bulletin fo Symbolic Logic 2 (1):1-71.

Kazakçi A, Hatchuel A (2009) Is "creative subject" of Brouwer a designer? -an Analysis of Intuitionistic Mathematics from the Viewpoint of C-K Design Theory? . In: International Conference on Engineering Design, ICED'09, Stanford CA, 24-27 August 2009, 2009.

Kazakçi A, Hatchuel A, Le Masson P, Weil B (2010) Simulation of Design reasoning based on C-K theory: a model and an example application. Paper presented at the International Design Conference - Design 2010, Dubrovnik - Croatia,

Kazakçi AO, Tsoukias A (2005) Extending the C-K design theory: a theoretical background for personal design assistants. Journal of Engineering Design 16 (4):399-411.

Kunen K (1980) Set Theory: an Introduction to Independence Proofs. Studies in Logic and The FOundations of Mathematics, 102. Elsevier,

Le Masson P, Hatchuel A, Weil B (2011) The Interplay Between Creativity issues and Design Theories: a new perspective for Design Management Studies? Creativity and Innovation Management 20 (4):217-237.

Mabogunje A, Leifer LJ (1997) Noun Phrases as Surrogates for Measuring Early Phases of the Mechanical Design Process. In: 9th international Conference on Design Theory and Methodology, American Society of Mechanical Engineers, September 14-17, Sacramento, CA, 1997. p 6

Poincaré H (2007) Science and Hypothesis. science and hypothesis was originally published in french in 1902 edn. Cosimo, New York

Reich Y (1995) A Critical Review of General Design Theory. Research in Engineering Design 7:1-18.

Reich Y, Hatchuel A, Shai O, Subrahmanian E (2010) A Theoretical Analysis of Creativity Methods in Engineering Design: Casting ASIT within C-K Theory Journal of Engineering Design:1-22.

Salustri FA (2005) Representing C-K Theory with an Action Logic. In: Proceedings ICED '05, Melbourne, Australia, 2005.

Schön DS (1990) The Design Process. In: Howard VA (ed) Varieties of Thinking. Essays from Harvard's Philosophy of Education Research Center. Routledge, New York, NY, pp 110-141

Shai O, Reich Y, Hatchuel A, Subrahmanian E (2009) Creativity Theories and Scientific Discovery: a Study of C-K Theory and Infused Design. In: International Conference on Engineering Design, ICED'09, 24-27 August 2009, Stanford CA, 2009.

Sharif Ullah AMM, Mamunur Rashid M, Tamaki Ji (2011) On some unique features of Ck theory of design. CIRP Journal of Manufacturing Science and Technology in press.

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

Worral J, Currie G (1980) Imre Lakatos, the methodology of scientific research programmes. Cambridge University Press, Cambridge

Yoshikawa H (1981) General Design Theory and a CAD System. In: Sata, Warman (eds) Man-Machine Communication in CAD/CAM, proceedings of the IFIP WG5.2-5.3 Working Conference 1980 (Tokyo). Amsterdam, North-Holland, pp 35-57

Yoshikawa H (1985) Design Theory for CAD/CAM integration. Annals of the CIRP 34 (1):173-178.

Zeng Y, Cheng GD (1991) On the logic of design. Design Studies 12 (3):137-141.

Zeng Y, Gu P (1999a) A science-based approach to product design theory: Part 1: formulation and formalization of design process. Robotics and Computer Integrated Manufacturing 15:331-339.

Zeng Y, Gu P (1999b) A science-based approach to product design theory: Part 2: formulation of design requirements and products. Robotics and Computer Integrated Manufacturing 15:341-352.