A new perspective for risk management: a study of the design of generic technology with a matroid model in C-K theory

Pascal Le Masson, Benoit Weil, Olga Kokshagina

Chapter xxA new perspective for risk management: a study of the design of generic technology with a matroid model in C-K theory[1]

Pascal Le Masson[1], Benoit Weil[1], Olga Kokshagina[1]

[1]MINES ParisTech, PSL University, Centre for Management Science, 60 Boulevard Saint Michel, 75 272 Paris Cedex 06, France; Tel: +33 1 40 51 92 21, E-mail: pascal.le_masson@mines-paristech.fr

Abstract

Risk management today has its main roots in decision theory paradigm (Friedman and Savage 1948). It consists in making the optimal choice between given possible decisions and probable states of nature. In this paper we extend this model to include a design capacity to deal with risk situations.

A design perspective leads to add a new action possibility in the model: to design a new alternative to deal with the probable states of nature. The new alternative design might also "create" new risks, so that a design perspective leads also to model the emergence of new risks as an exogenous "design process". Hence a design perspective raises two issues: can we design an alternative that would lower the risk? Does this new alternative create new risks?

We show (1) that minimizing known risks consists in designing an alternative whose success is independent from all the known risks – this alternative can be considered as a generic technology. We show (2)that the design of this generic technology depends on the structure of the unknown, ie the structure of the space generated by the concept of risk-free alternative. (3) We identify new strategies to deal with risks as dealing with the unknown.

Keywords:*Generic technology design, risk management, design theory, matroid, evolutionary model, structure of the unknown, robustness, decision theory, independence*

1.1 Introduction

Risk management is often seen as reducing the risk associated to a set of alternatives, be it technological risk or market risk. In *practice*, this raises critical issues since it requires to be able to list the risks and assess their probability of occurrence. Hence great debates on nuclear (risk assessment: the probability is very low but the consequences are so terrible that even low probability can not be neglected) or GMO (how to identify the type of risks), and more generally on "safety first" principle (also called precaution principle), which tends to favor "no action" alternative, since it is often the one implying the least (known) risks. Another issue appears in cases of "high uncertainty" or so called "unk unk", where the level of unknowness is so high that neither the level of risks nore even the list of risks are known. In R&D contexts, this corresponds to "double unknown" situations where neither technologies, nor markets are known. These situations are often considered as just unmanageable and they are left to gamblers and random processes. More *formally*, this approach of risk

[1]Unknowledgements: The Chair of Design Theory and Methods for Innovation, French Council for Energy.

management actually consists in applying models of *decision making* under uncertainty to select among a set of given designs and probable states of nature the one that maximizes the expected utility. Hence design is supposed to be already done; this approach does not directly consider design as a way to deal with risk. In this paper we would like show the paths opened by a design perspective on risk management.

On the one hand, a design perspective leads to add a new action possibility in the model: to design a new alternative to deal with the probable states of nature. On the other hand, the new alternative design might also "create" new risks, so that a design perspective leads also to model the emergence of new risks as an exogenous "design process". Hence a design perspective raises two issues: can we design an alternative that would lower the risk? Does this new alternative create new risks?

Main results: 1) we show that minimizing known risks consists in designing an alternative whose success is *independent* from all the known risks – this alternative can be considered as a generic technology. 2) we show that the design of this generic technology (or the design of the new independence) depends on the *structure of the unknown*, ie the structure of the space generated by the concept of risk-free alternative. 3) we identify new strategies to deal with risks and show that risk management in a design perspective shifts from dealing with uncertainty to dealing with the unknown.

1.2 Part 1 : setting the issue : beyond decision making, generic concepts to design low risk alternatives

1.2.1 Models for the interaction design / environment: the limits of decision paradigm

The notion of risk characterizes the consequences of an event for an actor. The event is considered as a "stroke of fortune" (and rather misfortune). If one adds some restrictive hypotheses on the structure of risks (these hypotheses will be discussed later), then a risk can be quantified as the product between the probability of the event and the consequences of this event for the actor.

More generally, without the restrictive hypothesis, one can consider risk as the relationship between, on the one hand, a "design space", where design uses, generates and transforms propositions and possibly leads to new artefacts; and, on the other hand, an "external" world, which can or shall *not* be transformed by design (it is "out of reach" of the design, it is *considered* as the states of nature that can't be changed), but which can interact (and even strongly) with the design. The "external world" is "invariant" by design [1]. But precisely for this reason, it plays a critical role: as an invariant, it strongly configures the design dynamic and strategies.

We can name many examples of such invariants: for instance, weather conditions: a design can be robust or sensitive to weather conditions, it will not influence the weather; norms, standards, design rules, consumer behavior, "production constraints", "process capabilities"… all are examples of such invariants in the external world. Risks are of this sort: for certain (many?) designs, terrorist attack, tsunami, earthquake, or just rain… are constraints of the external world that can't be influenced by these same designs but can influence them.

Two streams of research treat these kinds of constraints from the external world:

1- in a design perspective, a large stream of research, in particular in the US, has discussed the relationship between the design and its environment. In the 60s, Alexander proposed to consider design as the creation of a relationship between a "form" and a "context". Drawing the boarder between form and context *is* exactly the task of the designer, and, for Alexander, it can be considered as a dual process of problem setting and problem solving [2], which can be supported by more or less sophisticated and abstract patterns. Alexander proposes actually a dynamic process that leads to the

stabilization of a set of "specifications" that characterize the way the "context" is taken into account in the design space. Alexander actually opens two difference perspectives to deal with robustness, that are deepened in other approaches:

- On the one hand, the issue is to deal with a fixed set of functional requirements. More generally in many design theories and methods, "functions" or "functional requirements" precisely appear to play the same role than by Alexander: they represent the influence of the context on the design space (see Aximatic Design (AD)[3], General Design Theory (GDT)[4]). And the methods and theories deal with robustness by supporting the design of a solution that bot fit with the functional requirements and can support variations around this target. The design deals with a form of local invariance.

- On the other hand, the issue to "identify" the set of requirements that should be addressed by design. Some theories will focus on the way to identify this set. In Coupled Design Process (GDT)[5], the set of "requirements" is built during the process, since at each stage, the closure space of the tentative design leads to integrate new functions in the design. AD and GDT, can also be re-interpreted in this perspective: since the theories actually propose ways to address a given set of functions, as long as the design space follows some properties (Hausdorff space by GDT; axiom of independence between Functional Requirements (FR) and Design Parameters (DP) by AD), it means more generally that these theories offer ways to deal with complete functional spaces that follow these properties. In this case, robustness can be understood as the capacity to design a solution for large functional spaces, that follow specific properties. Or: if the "context" follows certain structural properties, then a design is possible. For instance in AD: if the "context" can be described a set of FR and if the FR and the DP follow the "independence" axiom, then a design is possible. The same for GDT: if the entity space is structured like a Hausdorff space, then a design is possible. This perspective explains that if the "invariant" –ie the context, like FR-DP and their relationship in AD- follows a certain structure, then the design is possible. This also holds for CDP and Infused Design (ID)[6,7]: CDP explains the effect of closure on the design; ID explain how to design by making use of structures in multiple domains.

To summarize : design theories and methods deal with robustness in two ways: either they address the issue of *local invariance* or they help to *address global invariance by adding specific conditions on the structure of invariance*; but they hardly study the interaction between varied structures of invariantce and the design space, ie the effect of certain structures of invariance on the design and conversely the sensitivity of design to certain structures of invariance.
Note that C-K theory can accept an invariance in K (see in particular [8]) and, contrary to other theories, does not make any hypothesis on the structure of invariance. This explains why it was possible to use C-K theory to study how the notion of "face force" emerged from an original "infused design" process in which the structures of two design domains didn't correspond exactly, the "hole" in the two invariant structure being the wellspring of an innovative design process. Hence it seems interesting to use C-K theory to study the consequence of certain invariance structures on design.

2-Decision-making perspective. The relationship between design and external environment has long been studied in a completely different stream of research, namely decision-making [9-12]. In decision-making, the question is to find a choice function that helps to identify the design alternative that best fits a set of probable states of nature. Contrary to the design- approaches, the set of states of nature can follow very general structures –it is only supposed to be a probability space-, just like the set of alternatives (which can be a simple list). Note that, even if this approach is not "design-oriented", decision-making largely used in R&D and engineering department to decide on the portfolio of projects to be launched. The "probable states of nature" are the markets and the decision alternatives are the products (or the technologies) to be developed. Another very useful case of application that uses the same framework are Taguchi methods, which actually help to analyse what is the best alternative to meet with varied external contexts.

Let's analyse the Wald statistical decision-making model (derived from decision theory), one of the most general models of decision-making under uncertainty. The models unfolds as follows:

- There are states of nature θ in Θ and random variables X_i within R^n with $L(X_1, X_2, \dots X_n, \theta)$ the likelihood function for $(X_1, X_2 \dots X_n)$ and $\mu(\theta)$, a priori density on θ. This is a representation of the states of nature, the probability of states is subjective and there is model associated to the subjective probability: the random variables model knowledge creation on these same states of nature - the a priori probability density on θ depends on X_i.
- There is a set of set of decisions δ (These are the known alternatives: known technologies, known products…)
- The relationship between the states of nature and the set of known decisions is modeled as follows: these is a cost function $C(\theta,d)$ that associates a certain cost to every pair (state of nature; decision) [this cost function can also be a utility function, see [9]]; the action consists in deciding for a certain $d_{\square}$, depending on θ, $\mu(\theta)$ and $\underline{x}_i$ (the results of the sampling operation). Ie one looks for a decision function ψ,

$$\underline{x} \in R^n \xrightarrow{\psi} \lambda_{\underline{x}}(\delta)$$

that leads to decide for a certain d. According to the decision-making theory, the best decision function, the function that minimizes the expected cost function:

$$E(C) = \int_{R^n} \int_{D} C(\theta, d)\, \lambda_{\underline{x}}(d)\, L(\underline{x}, \theta)\, \mu(\theta)\, d\theta\, d\underline{x}\, dd \qquad 1$$

Hence decision-making models helps to select the design alternative that has the best fit with a set of (subjectively) probable states of nature θ, $\mu(\theta)$ according to a cost function. This alternative is the "most robust" to the uncertainty on the states of nature. In this model, dealing with uncertainty does *not* consist in designing (the set of alternatives $\delta_{\square}$ is closed); it consists in acquiring knowledge to reduce the uncertainty on θ. Risk management is an uncertainty reduction process, not a design process. Still a research project can precisely be financed to reduce some technical uncertainty or, in marketing, a market research project can gain knowledge to reduce uncertainty on consumer behavior.

Hence the decision making perspective is very general and does not depend on invariance structures; but it enables only a limited form of action. Our goal in this paper is to cast this decision making approach into a design paradigm. We will see how this operation helps 1- to identify a new set of concept, namely "generic concept"; 2- some specific features of the design of generic concepts; 3- dynamic models with repeated interactions between "invariants" and related design space.

1.2.2 Casting the decision model into a design framework: the logic of generic concepts

To make a first step to cast decision making into a design perspective, let's rework the equation of the Waldian model. For sake of simplicity we consider that the sampling is reduced to 0 – ie there is no opportunity to gain more knowledge on the states of nature. Suppose that we can design an alternative d_{n+1} that would be better than all the other alternatives. It is easy to prove that the only property required by d_{n+1} is:

$$i = 1 \dots n, \quad \int C(\theta, d_{n+1})\, \mu(\theta)\, d\theta < \int C(\theta, d_i)\, \mu(\theta)\, d\theta \qquad 2$$

Or, without simplification:

$$i=1...n, \quad \int_{R^n} C(\ ,\ )\ {}_{\underline{x}}(d_{n+1})L(\underline{x},\ )\ (\ )d\underline{x}d\ <\ \int_{R^n} C(\ ,\ )\ {}_{\underline{x}}(d_i)L(\underline{x},\ )\ (\ )d\underline{x}d$$

In a design perspective, this equation actually is *the brief* (in C-K theory: the concept) *of a "robust" design*. This concept still depends on a priori probability μ. Actually such a d_{n+1} follwing equation 2 would be the best for all μ in the domain M:

$$M(d_{n+1})=\left\{\ /\ \ C(\ ,d_{n+1})\ (\ )d\ <\min_{i=1...n}\ C(\ ,d_i)\ (\ )d\ \right\}$$

An even more robust solution would be independent of μ. It means that whatever the belief on the states of nature – ie even for states of nature considered extremely low-, the alternative δ_{n+1} is the better. This can be written as:

$$i=1...n, \quad ,\ \ C(\ ,d_{n+1})\ (\ )d\ <\ \ C(\ ,d_i)\ (\ )d \qquad 3$$

Example: raincoat cap

Let's illustrate what it means on a simple example of decision making situation (see figure 1). Suppose that the decision maker wants to have a walk and his decision space is D = {d_1: take a cap to protect against the sun ; d_1, take a raincoat to protect against the rain}; the states of natures are Θ = {θ_1, sunny weather; θ_2, rainy weather}; and subjective probability are, for instance {$\mu(\theta_1)$ = 0,51; $\mu(\theta_2)$ = 1- $\mu(\theta_1)$ = 0,49}. The utility function is U $N^2 \rightarrow$R, for instance : U(θ_1; d_1)=100; U(θ_2; d_2)=10; U(θ_1, d_2) = 100; U(θ_2; d_1) = 10. This situation is usually represented by a decision-hazard tree (see figure 1). One computes the expected utility associated to each decision. With these data, the decision-maker should choose d1 – and one also understands the fragility of this choice, due to the proximity between $\mu(\theta_1)$ and $\mu(\theta_2)$. This remark usually leads to increase knowledge (eg. look at weather forecast even if it reduces the walk time).

If we add the hypothesis that the actor can design a new solution, then the dominating solution can be designed as d_3 such that U(θ_1; d_3)=100; U(θ_2; d_3)=100. The design of d_3 might lead to a kind of "raincoat-cap".

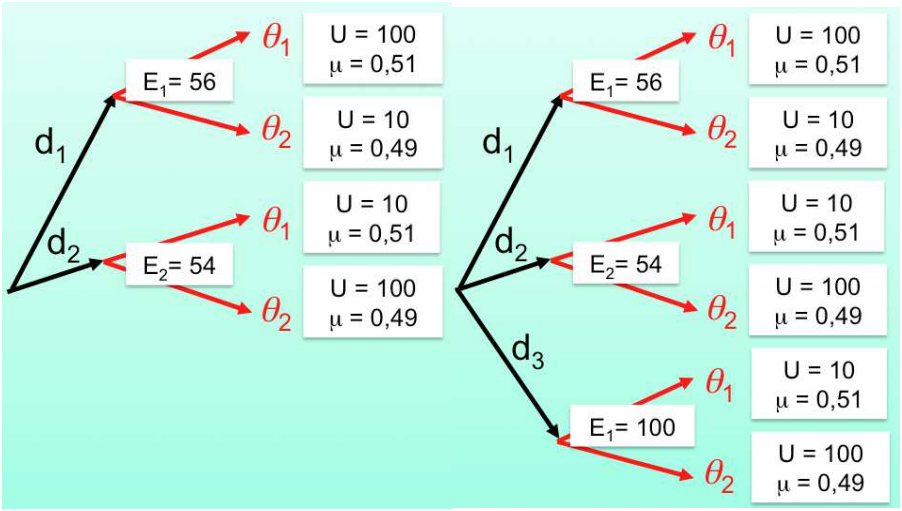


Figure 1: Raincoat-cap example

a) Selection of the solution with the best expected utility; b) design of a solution with dominating expected utility

Note that industrial history is actually full of such design. The graph in figure 2 illustrates the fact that some technologies are for instance *independent* of economics conditions. Even in recessions times, these technologies are

successful. As said Jean Schmitt, one famous VC in high tech industry: "some technologies do breakout even in breakdown times"

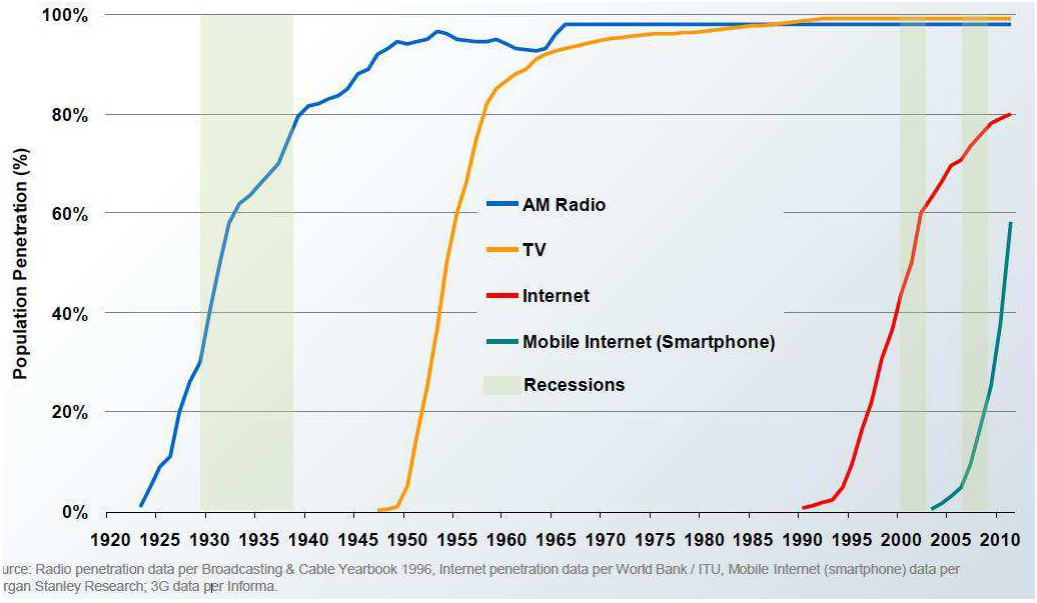


Figure 2: Technology adoption (measured by population penetration in %, in USA, Radio / TV / Internet / Mobile internet, 1920-2011. Source: Radio penetration data per Broadcasting & Cable Yearbook 1996, Internet penetration data per World Bank / ITU, Mobile Internet (smartphone) data per Morgan Stanley Research; 3G data per Informa

Comments: generic concept and new relationship "design space" / "context"

Let's underline some properties of the designed alternative:

1- **A conceptual alternative.** The nature of d_{n+1} is very different from all d_i, $i \leq n$: the latter are known alternatives whereas the former is a concept (in the sense of C-K theory: only a proposition that has no logical status). This is precisely because every d_i, $i \leq n$ are known that it is possible to evaluate for every d_i, i=1…n the costs or utility $U(\theta_j, d_i)$ for all $\theta \in \Theta$. By contrast, the only properties known on d_{n+1} are the one in the equation (3).

2- **Generic technology.** If such an alternative d_{n+1} exists, then this alternative is valid "whatever θ, whatever μ". Suppose that the states of nature are, for instance, varied markets and d_i are varied technologies to address –more or less well- these markets. Then d_{n+1} is a technology that addresses all the markets in Θ. This is a *generic technology*. These technologies are well-known in innovation economics [13]; but they are often studied ex post, when the technology is already designed and has already won on multiple markets [14]. The design perspective precisely enables to go one step further:

3- **Generic concept**. Since strictly speaking, d_{n+1} is not known but is a concept, we call it a *generic concept*. If this concept is designed, it would lead to a generic technology. *Hence equation (3) is actually the brief for any generic concept given a set of (subjectively) probable states of nature.*

Let's underline some consequences of this concept for the relationship between design space and "context" and for risk management:

1- **Invariance:** The new design space opened by d_{n+1} did *not* change the states of nature – they are still *invariants*.

2- **Independence:** But this new design space is made *independent* of these states. d_{n+1}*creates* a new *relationship* to the invariants.

3- **Risk management**: in this perspective, risk management does not consist in reducing uncertainty (even if this track remains open and the reduction of uncertainty also changes the design possibilities for d_{n+1}). We have already noticed that this design perspective does actually correspond to the design of generic technologies. But it would also be interesting in a lot of "controversies" situation (see the 20th century controversies on "smoking" or on "asbestos cancers" or today the debates on potential danger of electromagnetic waves created by wifi and mobile networks): usually these controversies are based on a logic of uncertainty reduction (hence debates on the "proofs" of "cigarette's cancer" or "asbestos's cancer"). In a design perspective, equation (3) leads to ask for the design of a concept like "as much pleasure as cigarette but independent of any risk of cigarette' cancer" or "as much fire protection as asbestos but independent of any risk of asbestos cancer". Note that it also opens a "design approach" for the "safety first" principle: the safety first principle requires that a technological alternative is *chosen* only if there is no risk; the default is that, with such a formulation, the principle can only be applied to "known" solutions. A design perspective of "safety first principle" actually leads the actor (for instance government, citizen associations…) to ask for the study of concepts that would follow equation (3).

4- **Creativity and system engineering synthesis**: the design of independence is actually at the root of a lot of engineering efforts. This is precisely what is required by the first axiom of Axiomatic Design or by Taguchi quality principles. It helps also to understand *a specific for of creativity in engineering design synthesis*: engineering design synthesis and creativity are often found contradictory – creativity brings a new dimension or a new technique that doesn't fit with existing systems and the synthesis rather consists in adapting the creative efforts to all system constraints. A generic concept is actually a creative path to deal with systems constraints by becoming independent of them! A generic concept does not add constraints but rather suppress (some of) them.

In a more dynamic perspective, the generic concept leads to study differently the states of nature:

1- **Long term stability**: being independent of a large set of states of nature, one can also consider that a generic technology is compatible with them. If over time the probability of states evolve (this time in a "natural way", not as a change in the subjective probability), the generic technology will remain dominating. In the case of raincoat-cap, even if global warming increases the probability of sun, the raincoat-cap remains the best solution. In case of technologies and markets, this means that a generic technology will survive many evolutions on the markets. Dynamically there will be a long term stability of a generic concept.

2- **Expansion in the states of nature**. On the other hand, the new technology will be sensitive to "new" states of nature, ie states of nature that were not in Θ, states of nature that were unknown – and not uncertain. It means that the generic concept actually "opens" a new set of risks that is strictly speaking unknown, that is *not* a combination of already known states, as if "nature" would be designing new states! Hence the design perspective also lead to *introduce a design logic in the regeneration of the states of nature*. For instance one can imagine that the raincoat-cap technology might be sensitive to electromagnetic waves. Hence the set of states of nature should be extended to include states like "electromagnetic storm". *Hence the risk emergence process should be considered as a design process, and even an expansive one.*

3- **Risk regeneration as an expansive design of the states of nature:** Note that this process of "expanding the states of nature" is not a modification of subjective probability, but it actually consists in (re)designing the probability space of the states of nature. The basic ground, the probability space Ω, A, P (where Ω is the sample space, A is the σ-algebra of events and P is a probability measure function – see Kolmogorov axiomatic) is extended to a new Ω'.

To conclude: the design perspective on risk leads to formulate specific concepts, that we call "generic concepts", which are of the form "there is an alternative that has a high utility whatever the states of nature taken in a set of states $\mathcal{O}$". Such a generic concept consists in designing an independence relationship with some invariants, namely the elements of Θ. In the next part we will a model of the design of such an independence. Based on this model, we will then study, in the last part, how this new design dynamically interacts with "invariances" that evolve over time.

1.3 Part 2: The design of generic concepts in matroids and algebraic extensions models

1.3.1 Beyond an evolutionary model of generic technology design: the example of Watt and Boulton reciprocating steam engine

Having identified a generic concept, we are interested in studying the design of such a concept. There are already implicit models of the design of generic technologies. The usual one is an "evolutionary, random" model: a generic technology is a randomly emerging design that is applied progressively on a sequence of applications, and application after application this design appears dominating on a (large) subset of these applications. The story of the steam engine is often told this way: Watt designed a steam engine and progressively many applications were found for it.

Still the evolutionary model, where one "species" progressively adapt to multiple environments, is only one possibility. As shown in [14,15], this model actually does *not* correspond to how Watt and Boulton historically designed steam engine as a generic technology: based on historical books, Kokshagina et al show that there was already steam engines, and they were adapted to mining, but not to other uses; hence steam engine in the 70s was not a generic technology (see illustration in figure 3). In the 1780s, Boulton asked Watt to work on a generic concept "a steam engine that is compatible with multiple machine tools"; and Watt designed a specific "steam-engine technology" for this concept. Surprisingly enough, it even appears that the key issue was very specifically to design a new way to transmit movement from steam engine, namely a "reciprocating steam engine". Hence a strange paradox of the design of generic technology: it seems to be a "complete" original technology (steam engine), but its design actually focuses on a detail in the complex system (transformation of the movement of the steam engine rod).

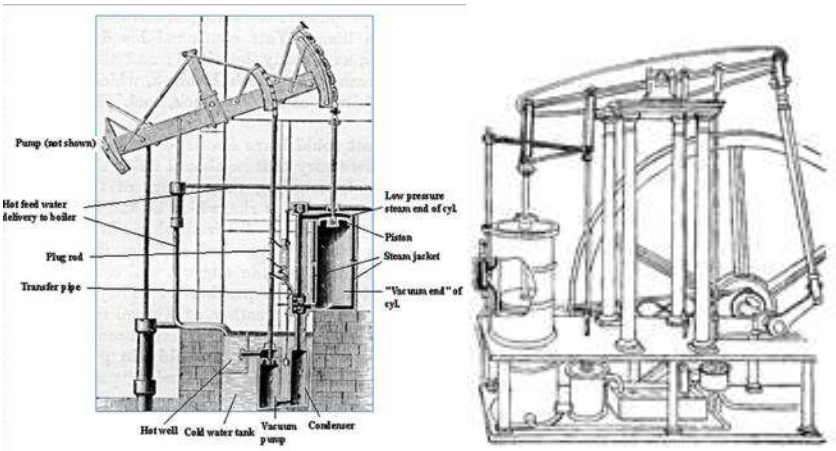


Figure 3: The design of steam engine as a generic technology, by Watt and Boulton

a) 1763 Watt steam engine with separate condensation chamber; b) 1784 Watt & Boulton Double acting steam engine (parallel motion or so-called "reciprocating steam engine").

This historical case underlines that the evolutionary model might hide more intentional and complex processes, and there might be a variety of processes to design a generic technology.

1.3.2.The issue: designing "whatever theta"

The issue is to understand the specificities of a design process that designs one independence between the design space and some invariances. The evolutionary model deals with it in a sequential process: a technology d_0 is designed for one θ_0, and then its utility is tested on each of the other elements $\theta_{\square}$ of Θ. But suppose that Θ is made only of linear variations of θ_0, ie every $\theta_{\square}$ is of the form $\theta_{\square} = a_i . \theta_0$ where $a_i \in R$; then it not necessary to test d_0 for each $\theta_{\square}$ but it is enough to test (or redesign) for the *operation* "multiplication of $\theta_{\square}$ with a real number". Hence the design of generic technology takes advantage of the *structure* of the invariances (here linear dependence in Θ). And it finally designs a specific relationship –hence a structure- between the invariances and the design space.

Hence to study the design of generic technology, we need a model of knowledge structures and their evolution during the design process. As mentioned earlier, C-K theory can be useful in our case, because: 1- invariance is possible in C-K; 2- the knowledge space is a "free" variable, in the sense as C-K theory is supposed to work with many models of K. What we need to study the design of generic technology with C-K is just to add a specific model of K.

In this paper we choose to consider that pieces of knowledge can be studied as matroid structures. Why matroid? Because matroid is a very general language to deal with independence in many equivalent models (graphs, linear algebra, field extensions,…). Moreover, it offers ways to characterize structures and their evolutions (a structure can be characterized by rank, circuits, bases, lattice of flats; the evolution of structures can be modeled with operations of duals, minors, sums, deletion, contraction, extension,…). Hence adding matroid structures to K-space (in C-K theory) provides us with powerful analytical tools to follow the transformation of structures during a process of designing a generic technology. Note that in matroid theory, the operations on matroids were mainly used to "analyse" matroids, ie to identify micro-structures into more complex ones. We will use the same tools to rather understand how, during a design process, new structures emerge from given ones.

1.3.3. A model of C-K with matroids in K

Matroid structures were introduced by Whitney, in the 30s, to capture abstractly the essence of (linear) dependence. A matroid is a pair (E, I) consisting of a finite set E and a collection I of subset of E satisfying the following properties: i) I is non-empty; ii) every subset of every member of I is also in I (I is hereditary) ; iii) if X and Y are in I and $|X|=|Y|+1$ (the operator $|\ldots|$ designates the number of elements in a set of elements), then there is an element x in X-Y such that $Y \cup \{x\}$ is in I (independence augmentation condition). I are the independent sets of a matroid on E, M(E). There are many forms for matroid (defined on matrices, on algebraic extensions,…). In particular, it is very easy to consider the matroid given by a graph: Given a graph G with vertice V(G), the set of vertices of the graph and E(G) the set of edges of the graph. Then let I be the collection of subset of E that do not contain all of the edges of any cycle closed path (or cycle) of G. Then (E, I) is a matroid on G; it is called the cycle matroid of the graph G and is noted (M(G).

We use this structure to model the design process associated to the design of generic technologies:

1- In K: We consider that K contains a graph G. One interpretation of this graph can be as follows: the vertices are some functions fi; an edge represents a technology to address a pair of functions. A path of edges defines a technology (a

combination of technologies) to address all the vertices in the path. The graph G below can be interpreted as a synthesis of the technological know-how of the designer. For instance in the figure 4 the designer knows how to address $\{f_4; f_5\}$ (with the edge e_{45}); and he knows several solutions to address $\{f_1; f_2; f_3\}$ (e_{12}-e_{23} or e_{13}-e_{23}); he doesn't know any solution to address $\{f_5; f_6\}$ or $\{f_3; f_5\}$. This graph of designer's knowledge is a matroid.

Note that this matroid can also be characterized by its basis or its cycles. It has a certain rank, which actually corresponds to the size of the largest independent set. In a graph G, we have: rank(G) = |V(G)|-ncc(G) where |V(G)| is the number of vertices in G and ncc(G) is the number of connected components in G. (rank(G)=4 in the example below). The rank illustrates the most "complex" technologies that can be built by the a user of the K-space: at most the graph G below enables to separately address four independent edges.

The matroid also has so-called "flats". A flat is a set of elements such that it is impossible to add a new element to it without changing its ranks. One says that a flat is a "closed" set in a matroid. In G below, $\{e_{12}, e_{23}, e_{13}\}$ is a flat and $\{e_{12}, e_{13}\}$ is not a flat. The flat represent stable "substructures" in a matroid.

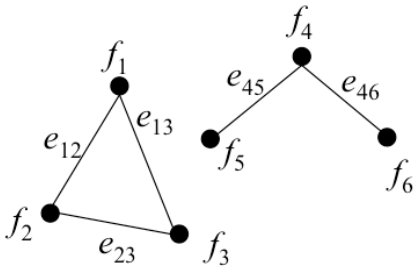


Figure 4: A graph G

We also represent in K the context invariance, representing all the states of nature that the design could be facing. We consider for instance that the designer might need to address any combination of functions taken from the set of functions $F=\{f_1; f_2, \ldots f_n\}$. And we consider (for sake of simplicity) that any market is independent of the other (the market $\{f_1; f_2\}$ is independent of $\{f_1; f_2; f_3;\}$ even if the functions of the former are included in the functions of the latter). This context invariance is also a matroid: it is one of the so-called "uniform matroids" $U_{p,n}$ where the elements are the n-first integers and the independence sets are all subsets of E of size equal or less than p. The contexts can hence be represented by a matroid $U_{n,n}$, in which all the subsets of E are independent. With these two matroids we have the structures of the known, in K space.

2- In C: In the model, designing the generic concept consists in designing additional edges in G to address all the markets in $U_{n,n}$. It actually means to design the missing edges so that the graph built on F becomes complete – ie: each vertex is linked to all the others by one single edge. This complete graph is called K_{n-1} in matroid theory. It means that if F=V(G) and G is complete, then the designer knowledge is a generic technology for $U_{n,n}$ built on F.

In C, it is possible to represent the graph of all the missing and required edges: from the graph G, one represents all the missing and required edges ; one contracts all the edges that exists and are necessary; and one skip all the edges that are known. If $F=\{f_1; f_2, \ldots f_6\}$, this gives the C-graph in figure 5. Note that this C-graph is also a matroid. It is a connected matroid, and, deleting the loops, it is even a complete graph. Its rank equals the number of connected components in K.

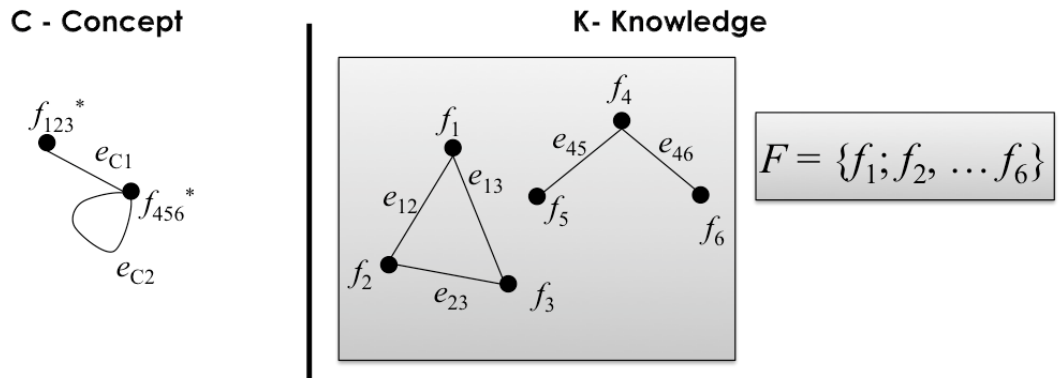


Figure 5: C-K structures with matroids in K.

Let's see how the design of the missing edges occurs. There are two different processes to add one single edge, and the matroid model will help us to understand their critical features:

1- **Constant-rank extension:** this concerns the loop-edges in C. the graph G can be "completed" with a new edge following a (single) *extension* process that does *not require changing the rank of the matroid*. It means that the new edge does not add a level of sophistication, a new dimension to the graph. It can be proven (see [16]) that any single extension of this kind corresponds to a flat whose rank will be unchanged by the new edge. Such a flat is called a modular cut. In graph G above, the edge e_{56} linking f_5 and f_6 can be added without changing the rank of the matroid; and it corresponds to the modular cut $\{e_{45}; e_{46}\}$.

   It can be shown that this process can be repeated until it completes all the connected components.

   The constant rank extension is a process to create new *dependent* sets. For instance the new edge e_{56} creates also $\{e_{56}; e_{45}; e_{46}\}$. More generally, the structure of the dependent sets can be measured by the rank of the dual of the graph, also called the corank. If the graph is a spanning tree on n vertices, the rank is n-1 and the corank is zero (no dependent set in the graph). If the spanning tree is completed following the constant-rank extension, its becomes a complete graph K_{n-1}, the rank is kept to n-1 but the corank will reach (n-1)(n-2)/2.

   This description corresponds to the fact that for a set of functions $\{f_4; f_5; f_6;\}$, all the associated markets will be addressed and for the most sophisticated markets involving more than 2 functions, there will be several technological combinations to reach it. The constant-rank extension hence helps to address all the markets and, simultaneously, to offer multiple technological alternatives to address each markets. This actually corresponds to a very powerful form of robustness.

2- **Rank-increase extension:** The constant rank extension is not enough to address all situations of generic technologydesign: the issue rises when the set *F* contains vertices from different connected components in G. It is then required to design a new edge. This new edge *e* has very different properties as the edge designed by constant-rank extension: every flat of *G* will change rank when *e* is added and *no new dependent set is created*. This extension process hence *adds one dimension* to the graph [16]. For this reason, this can be assimilated to an expansive partition.

   Such an edge corresponds to the fact that two distinct sets of functions that were not connected at all will now be connected by one single new edge. This opens the possibility for more complex technologies. Note that adding such an edge always means that the new "technology" is *compatible* with all already existing technologies (since it enables

all new paths that use the new edge and other existing edges). A good example of such an "edge" is Watt and Boulton "reciprocating movement" that enabled to link the steam engines technologies (one connected component) to the machine tools technologies (another connected component)

The complete design process will (at least) then proceed as follows:

1- Constant-rank extension until all connected components are complete.

2- Connect connected components. There are several possibilities here, since the C-graph is complete and only a spanning tree is required. Suppose that all the C-edgehave a certain cost: then the problem consists in finding the spanning tree of minimal weight over a matroid structure. This can be done with the Greedy algorithm (Kruskal algorithm in the case of graphs).

3- The resulting graph in K is now connected but is not complete. It is necessary to identify in C the missing edges and to proceed with anew with a constant rank extension. Some properties of this third step should be underlined:

   a. The new edges that will be created are necessarily associated to a flat that contains at least one edge resulting from an expansion process. In this sense all the new edges are now the consequence of the expansive edges.

   b. But on the other hand, each new edge of the third step creates a new circuit that includes an expansive edge. This means that it also creates *substitute* for the expansive edge.

Note that this process is not deterministic and keep the "generative" aspect of design: for instance there are many possibilities for the spanning tree, depending on the weight that the designer will put on all the C-edges. For instance the weight can be based on the estimated cost of the technology development; but it can also be linked to the expected difficulty to make this new edge compatible with already existing edges in the two distinct connected components,… Many other weight systems are possible – note that whatever the weights chosen, the matroid structure of the C-graphs warranties convergence of Greedy algorithm.

We say that the process will *at least* go through the three steps: actually additional steps are possible. For instance, the designer might himself add a new "function" to be addressed – this is far from unsual: it would just means that the designers add a constraint. Of course this enw function should be then be handled to keep independence.

1.3.4. Main properties for the design of generic technologies

The model with matroid helps to understand **critical properties on the design of generic technology (GT)**: GT results from two "genericity building" operations: G1 (in step 2) connects by preserving (necessary) past connections; and G2 (in steps 1 and 3) completes the graph, ie creates new dependent structures in G based on the first new expansive edge. These two processes explain apparent paradoxes in the design of GT:

1- Because of G1: a generic technology includes both the longest cycle, hence the "most constrained" one, *and* a local property (just one missing edge). This corresponds to the logic of genericity in steam engine, where genericity was made by working on reciprocating movement to link the technologies of steam engine with the technologies of machine tools. More generally, the design of generic technology consists in designing to creates new compatibilities between "islands" of disconnected technologies.

2- Because of G2: every new edge is a consequence of the first expansive edge (hence a function of it) but, in the end, every new edge is also a possible substitute for the initial expansive edge. Hence a generic technology also encloses variances and alternatives. It is not one single solution but rather a set of interdependent solutions.

3- **Discussing the evolutionary model of GT design:** We can easily represent in the matroid model the evolutionary process mentioned above for the design of GT: it consists in building new edges for each single new market. It is interesting to note that this kind of random, evolutionary process can certainly help to complete a connected graph but will hardly manage to create the edge that connect two separate connected components: of course one market might require to link to vertices from disconnected components, but this market would hence allow for a technology that is not necessarily compatible with all already existing technologies in the connected component. This only possible for such a connection is a market that needs all the function of the two disconnected components. *Hence a random, evolutionary process is quite unlikely to lead to a generic technology; it should at least be guided by the requirement to connect disconnected components.*

The model also enlightens some aspects of **"independence"** in generic technologies:

1- the design of GT actually establishes a certain structure between designed world and environment. Here this independence is actually modeled as a complete graph K_{n-1}. One can underline that in terms of matroids, this structure has interesting properties: K_{n-1} and $U_{n,n}$ have the same rank; one can even notice that their lattice of flat are isomorph (see [16]) ; but they have very different corank, 0 zero for $U_{n,n}$ and (n-1)(n-2)/2 for K_{n-1} and hence very different dependent sets. *This means that a generic technology appears as a set of compatible, more or less substitutable technologies.* In a sense the knowledge set of an engineering department should embody a generic technology!

2- Note that the design process actually works on several structures: two structures in K, modeling what is known; and also a structure in C. The latter is interesting: this is **the structure of the unknown!** This structure represents all the "holes" in K and their interdependences. The design consists in building this structure and "extracting" some edges from it to make them become knowledge. It can be noted that the C-structure follows a matroid structure which enables a relatively easy process to identify a spanning tree in the C-graph.

1.4 Part 3: Dynamic models of the interaction designed world/environment

Based on the model above, we can now study the dynamic interaction between a designed world and some "constraints".

1.4.1. Dynamic model

As already noticed in part 1, the logic of generic technology leads to a new logic to follow the dynamics of context. As long as a new "context" is actually a dependent set in the structure of external structures, then this context is taken into account by the generic technology. Hence the only change in context that would lead to a new design are context that include a property that is independent from the past *F*. It can be a new function f_{n+1}.

In a dynamic process, this kind of extension might lead to several forms:

1- if the successive sets of risks F(t) are included one after the other (for each t, $F(t) \subset F(t+1)$) , then the process leads at each time t to a complete matroid K(t) with $K(t) \subset F(t+1)$

2- But there is another possibility: suppose that the set F(t+1) contains a new function f_{t+1} but does not contain all previous functions. Then in C, edges that are known and not necessary are deleted. By decreasing the requests for compatibility, this simplifies the task to link the new function to the past one. Back in K, this leads to a new connected component associated to the set F(t+1) but also to other connected components associated to the functions that were connected earlier but are not more necessary for F(t+1) (see figure 6)

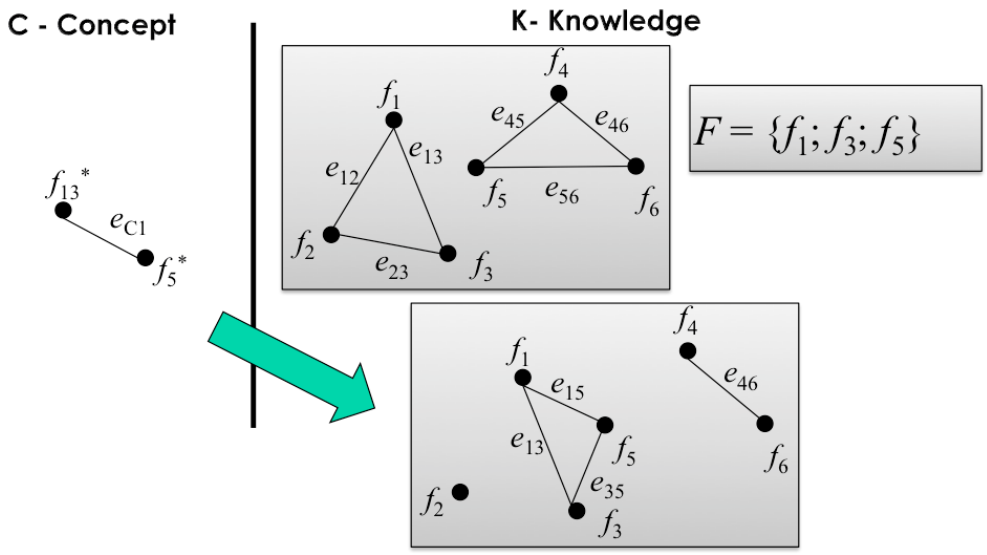


Figure 6: Dynamic models of interaction designed world / context

This second process actually leads to the emergence of complex but separate connected components in K.

1.4.2. Using the model to interpret some dynamic situations

1) A first situation to analyze is the **"imperfect design"** of generic technology: in part 2, we made the hypothesis that the designer would be able to address all the edges required. Suppose now that there is a budget limit. Then it is possible that the budget is not sufficient to create a spanning tree for the C-graph. Or the designer will select a subset of F to get only a partial generacity. We say that the design of GT is imperfect. Interestingly enough, these behaviors can be represented in the model above. They all fall in the second situation, either because the set F is restricted in advance, or because the new edges to be designed can only address a subset of F., due to cost reasons. It explains how over time, one can have the stabilization of a connected subgraph without relation with active parametres of F.
2) The dynamic model also leads to come back to evolutionary models in a richer way: instead of considering a selection process based on one given state of nature, one can represent the evolutions of probable states of nature, ie as subsets of F. In this extended evolutionary model, one can find *partial genericity* changing over time and separate connected subgraphs that can suddenly appear useful if, at a certain time t+q, F(t+q) suddenly reuses some of the "old" functions and hence the connected component becomes relevant again.

1.4.3. One application: evolutionary models in biology

The dynamic model of the design of generic technologies might be relevant for some technological evolutions; but one can underline that actually this model might even be relevant for models of evolutionary process in biology.

Suppose that one represent a species as a designer who master certain "technologies" to address external conditions; suppose that these conditions evolve over time but the species never completely adapt to one context but keep a "robust" strategy by adapting to several contexts. The species actually acts as a designer of a (partially) generic technology. What does our model help to explain:

- the model is coherent with stasis and punctuated equilibria: if the context evolves inside the set of alternatives addressed by the species, the "technologies" are not changed; the species evolve only when an "original" function, out of the robustness scope, appears.
- the model is coherent with **exaptation situations**[17]: exaptation is a situation where an organ (a technology) is developed to adapt to a situation –the panda's thumb was developed to go up and run in trees at a time where panda were smaller and lived in a dense forest context- ; then this function is less needed but the organ remains and the species go on evolving, keeping the original trait – the panda doesn't not need to go up and run in trees but changes in his environment lead him step by step to eat bambous- ; and finally the function holding a branch" become important again but this time to eat bamboo: the panda's thumb becomes important again.

Hence our dynamic model could account for contemporary forms of complex, evolutionary processes in biology!

1.4.4. One application: evolutionary models in biology

We already underlined (part 2) how our model could help to design generic technology (constant-rank extensions and rank-increase extensions). Some lessons can also be learnt from the dynamic model: it appears that the critical issue is the emergence of the new function. In part 1 we considered that this new function could be interpreted as a form of "expansive design" in nature. In a risk management perspective, it is also important to manage the emergence of the new function. It consists in launching "exploratory projects" that help to extend the list of "risks" (or F in our model).

The dynamic model helps to understand why such an exploration can just work! Apparently, such an exploration is very difficult, since "nature" doesn't speak in advance and the new function can be everywhere! Still the dynamic model shows what should be explored in such an exploration: 1) the risks that are *not* in F. In a sense it restricts the exploration to brand new functions, it leads to better characterize F and its structure to understand what is out of F and its structure; 2) the risks are impacting the known technologies, usually in a negative way since it proves that it is not robust to a new F.

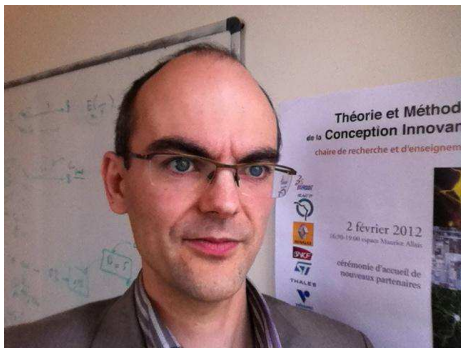
Note that it explains why this kind of exploration was called "crazy concept" exploration: it consists in exploring a concept that is "out" of the usual set of risks (or market opportunity) and that can not be addressed with available knowledge!

1.5 Conclusion

In this paper we study generic concepts, ie concepts for artefacts that would be valid for a large domain, ie for a large set of external conditions. We have shown that these concepts can be derived from decision making theory and are of the form given by equation 3. Equation 3 means that such a generic concept establishes an "independence" with the set of possible contexts. We have then analyzed the design based on a generic concept. We used a model derived from C-K theory with matroids in K-space. We showed that the design process combines two different operations, constant-rank extension and rank-increase extension and these properties explain critical properties of generic technologies as well as the limits of evolutionary processes to design generic technologies. Finally we study dynamic model for the design of generic technology and show that these dynamic models account for "imperfect design" and could open new perspective for the study of classical "risk management" in biology; the study of evolutionary processes.

Pascal Le Masson is Professor at MINES ParisTech, Chair of Design Theory and Methods for Innovation. He is the Director of the Center for Management Science. His research unfolds in three main directions: i/ design theory (C-K theory,

mathematical foundations,…) ii/ collective innovative design methods (creativity, prototyping, user involvement processes) iii./ innovative design organization (advanced R&D, iv/ economics of design.He has published several papers and a book “Strategic Management of Innovation and Design” (co-authored by Armand Hatchuel and Benoit Weil, Cambridge University Press). He co-chairs (with Eswaran Subrahmanian and Yoram Reich) the “Design Theory” Special Interest group of the Design Society.



Benoit Weil isProfessor at MINES ParisTech, Chair of Design Theory and Methods for Innovation. His research areas are 1) Design Theory and models for Design Science; 2) Design and R&D Management; 3) Management of the Innovative Firm; 4) Theory of Design Regimes; 5) R&D and Design history. He has published several papers and a book “Strategic Management of Innovation and Design” (co-authored by Armand Hatchuel and Pascal Le Masson), Cambridge University Press).

Olga Kokshagina is PhD candidate at Center for Management Science, Mines ParisTech School, France. Her research focuses on the management of innovative design capabilities and more particularly on the areas of innovation and technology management, projects portfolio management, uncertainty management and R&D tools and methods. Her thesis dissertation models the design of generic technologies and investigates the associated models of collective action. Being interested in the specifics of innovative processes in high-tech industries such as nano-, biotechnologies, semiconductors, IT, telecommunication, she has conducted her dissertation work in collaboration with a leading European semiconductor company - STMicroelectronics.

References

[1] Armand Hatchuel, Benoît Weil, and Pascal Le Masson, "Towards an ontology of design: lessons from C-K Design theory and Forcing," *Research in Engineering Design* 24, no. 2 (2013): 147-163.
[2] Christopher Alexander, *Notes on the Synthesis of Form*, 15th printing, 1999 ed. (Cambridge, MA: Harvard University Press, 1964).
[3] Nam P. Suh, *Principles of Design* (New York: Oxford University Press, 1990).
[4] H. Yoshikawa, "General Design Theory and a CAD System," in *Man-Machine Communication in CAD/CAM, proceedings of the IFIP WG5.2-5.3 Working Conference 1980 (Tokyo)*, ed. T. Sata and E. Warman (Amsterdam, North-Holland: 1981), 35-57.
[5] D. Braha and Y. Reich, "Topologial structures for modelling engineering design processes," *Research in Engineering Design* 14, no. 4 (2003): 185-199.
[6] Offer Shai and Yoram Reich, "Infused Design: I Theory," *Research in Engineering Design* 15, no. 2 (2004): 93-107.
[7] Offer Shai and Yoram Reich, "Infused Design: II Practice," *Research in Engineering Design* 15, no. 2 (2004): 108-121.
[8] Armand Hatchuel and Benoît Weil, "Design as Forcing: deepening the foundations of C-K theory," in *International Conference on Engineering Design* (Paris: 2007), 12.
[9] Leonard J. Savage, *The foundations of statistics*, 2nd edition (1st edition: 1954) ed. (New York: Dover, 1972).
[10] Milton Friedman and Leonard J. Savage, "The Utility Analysis of Choices Involving Risk," *Journal of Political Economy* 56, no. 4 (1948): 279-304.
[11] Howard Raïffa, *Decision Analysis* (Reading, MA: Addison-Wesley, 1968).
[12] Abraham Wald, *Statistical Decision Functions* (New York: John Wiley & Sons, 1950).
[13] Timothy F. Bresnahan and Manuel Trajtenberg, "General Purpose Technologies: Engines of Growth?," *Journal of Econometrics* 65, no. 1 (1995): 83-108.
[14] Olga Kokshagina, Pascal Le Masson, and Benoît Weil, "How design theories enable the design of generic technologies: notion of generic concepts and Genericity building operators " in *International Conference on Engineering Design, ICED'13* (Séoul, Korea: 2013).
[15] Olga Kokshagina and others, "Platform emergence in double unknown: common challenge strategy," in *R&D Management Conference* (Grenoble, France: 2012), 25.

[16] James Oxley, *Matroid Theory*, ed. R. Cohen, et al., 2nd edition ed., Oxford Graduate Texts in Mathematics (Oxford University Press, 2011).
[17] Stephen Jay Gould, "The Panda's Thumb of Technology," *Natural History* January 1987 (1987).