

Designing techniques for systemic impact: lessons from C-K theory and matroid structures

Pascal Le Masson¹  · Armand Hatchuel¹ · Olga Kokshagina¹ · Benoit Weil¹

Received: 20 August 2015 / Revised: 24 September 2016 / Accepted: 28 September 2016 / Published online: 24 October 2016
© Springer-Verlag London 2016

Abstract As underlined in Arthur’s book “the nature of technology”, we are very knowledgeable on the design of objects, services or technical systems, but we don’t know much on the dynamics of technologies. Still contemporary innovation often consists in designing techniques with systemic impact. They are *pervasive*—both invasive and perturbing—and they recompose the family of techniques. Can we model the impact and the design of such techniques? More specifically: how can one design generic technology, i.e. a single technology that provokes a complete reordering of families of techniques? Advances in design theories open new possibilities to answer these questions. In this paper, we use C-K design theory and a matroid-based model of the set of techniques to propose a new model (C-K/Ma) of the dynamics of techniques, accounting for the design of generic technologies. We show that: (1) C-K/Ma accounts for basic phenomena in the design of pervasive (and non-pervasive) techniques, in particular for generic techniques. (2) C-K/Ma, when applied iteratively, helps to propose new laws for the dynamics of techniques and helps to build strategic alternatives in the design of techniques. Moreover, C-K/Ma contributes to design theory since it provides some basic quantifiers and operations that could lead to a computational model of the process of designing techniques with systemic impact.

Keywords Design theory · Independence · Generic technology

✉ Pascal Le Masson
pascal.le_masson@mines-paristech.fr

¹ CGS-i3, CNRS, UMR 9217, MINES Paristech, PSL Research University, Paris, France

1 Introduction: designing for systemic impact?

In his book “the nature of technology” (Arthur 2009), W. Brian Arthur explains that looking for “some common logic that would structure technology and determine its ways and progress” he couldn’t find it. Is this claim exaggerated in the light of the literature in the engineering design community? Not that much: we are very knowledgeable on the design of objects, services or technical systems. We are used to relying on elementary techniques to design them. But what do we know on the design of these “techniques”? What do we know about the dynamics of techniques?

This question is all the more relevant today because the dynamics itself might be strongly changing. Let’s consider some famous technologies and their impact: software in mechanical systems (aeronautics, mechatronics), semiconductors in a large variety of systems, additive processes (or 3D printing) in industrial processes, etc. All these technologies have a “systemic” impact, they are *pervasive*—both invasive and perturbing—and they recompose large parts of the set of existing techniques. We know of some technologies that completely changed the technical environment of their time—“steam engine”, “electricity”, etc. One even associated each industrial revolution to a handful of such techniques. Today this process of *reorganizing* the family of techniques might be much more frequent. And it is not sure that we understand it: what is the origin of such pervasive techniques? How can we model their impact? Are there design strategies to design the techniques and their impact? And even more specifically: how can one design generic technology, i.e. a single technology that provokes a complete reordering of families of techniques?

We tend to think about these dynamics of techniques in terms of “combination” and “assembly”, maybe stuck in

the “mechanical” paradigm. Or we rely on evolutionary models, with random emergence and “natural” selection. Some authors have tried to identify laws of the evolution of techniques. These proposals contributed to enlighten some facets of the dynamics of techniques but also raised difficult questions: why two independent domains of techniques become suddenly combinable? Can we differentiate between a “local” combination and a more “generic” one? Why some techniques are just locally solving a problem while other might lead to generate entire lineages of descendants?

To answer these questions, we actually lack of models. We today understand how a new “individual” emerges in the order of techniques [the logic of “individuation” analysed by (Simondon 1958)]—or we understand how the order of techniques evolves—and occasionally gets stuck [see evolutionary models or the “blocked technical systems” of (Gilles 1986)]. But we also need to overcome this separation between “individual” and “population”: we need to model the design of a new “individual” with a specific effect on the “population”, the design of technique with a specific effect on the set of techniques. Such a model of the design of a new technique in the set of techniques should more specifically account for critical phenomena in the design of techniques:

1. It should account for contrasted forms of design: (a) in many situations it is asked to design one product by relying on techniques, *without* designing a technique; (b) in engineering department, it often happens that designers design one new technique to develop one specific product and it asked that this new technique does not provoke a complete change in the set of techniques that this new technique is *as least pervasive as possible*; (c) by contrast, some inventors are famous for having designed techniques that provoke a complete reordering of techniques, that are strongly pervasive, that are generic and connect suddenly applications and industrial sectors that were previously unconnected. A model of the design of techniques should account for the design of non-pervasive *and* pervasive techniques.
2. It should lead to discuss basic models of the dynamic of techniques and even propose some basic laws for this. In particular, the model should help enrich basic models of “combinations” to include the logic of endogenous creation of techniques. In particular, the model should help to account for repeated design of techniques—how, over time, repeated designs interact with and transform the structure of the set of techniques.

In this paper, we tackle one specific issue for such models: our basic assumption is that modelling the design

of a technique in a set of techniques should be based on modelling the interdependencies between the techniques. Understanding how one new technique is pervasive or not consists in understanding how it creates interdependencies where there were independencies and how it creates independencies where there was mutual conditioning. A new technique changes the set of techniques (or not) by changing the structure of interdependencies (or not).

Still any specific model of relations between techniques will have strong limitations. If one models techniques as graphs or as a vector in a vectorial space (as actually Suh does), one limits the validity of the model. The basic idea of this paper consists in relying on a “pure” model of interdependencies. We show that a model based on matroid theory is adapted for this (Le Masson et al. 2015a) (part 2). Moreover, recent advances in design theories open new possibilities to use a specific “knowledge model” and analyse its evolution by design (Kazakçi et al. 2010). Hence the paper proposes a model of technique design based on design theory (more precisely C-K theory) and matroid theory, a model called C-K/Ma. We show that:

1. C-K/Ma accounts for basic phenomena in the design of pervasive and non-pervasive techniques, in particular for generic techniques (part 3).
2. C-K/Ma, when applied iteratively, helps to build quantifiers, that enable to measure this impact of a new technique on a set of interdependent technique and helps to propose new laws for the dynamics of techniques and helps to build strategic alternatives in the design of techniques (part 4).

Moreover C-K/Ma contributes to design theory since it provides some basic quantifiers and operations that could lead to a computational model of the process of designing techniques with systemic impact.

2 Research questions: interdependences and the design of the system of techniques

2.1 Endogenous dynamics of technologies: the design of interdependencies

In the multiple approaches on the dynamics of technologies, we can distinguish two main trends.

1. A *first* stream of works tends to consider that the dynamics of technologies is based on the invention of new functional means. Arthur’s book provides a synthesis of these approaches (Arthur 2009). Technologies are functional building blocks. The scientific study of phenomena regularly provides new building blocks. They are “*combined*” into artefacts.

Combinations are selected—for instance by the markets. Hence, the dynamics of technologies are controlled by market and science. This kind of model is used in many evolutionary economics works (Dosi et al. 1988; Saviotti and Metcalfe 1991). This approach is “exogenous”: the dynamics of technologies mainly relies on exogenous forces—market and science. In this perspective, one tends to define a technique as a “functional” building block (e.g. Arthur considers that a technique is “a means to fulfil a human needs or purpose”), without considering the “combinations” issues. In this first approach, one tends to consider techniques as Lego blocks and one invention is the design of new kind of block. In the world of Lego, each new Lego block is designed to be compatible with all previously known blocks. This approach has some limits when it comes to situations where techniques are only partially compatible with each other.

As a consequence, this approach tends to neglect the issue of the design and evolution of the combinative capabilities of techniques. In particular, in this perspective, it is more difficult to analyse the emergence of techniques whose main property is precisely the capacity to assure the compatibility *between* existing techniques. Let’s mention two cases: (1) when Watt and Boulton invent the cinematic for a rotary steam engine, they actually “just” make compatible the steam engine and the world of machine tools. This technique has a genericity property: it increases the genericity of the steam engine (now compatible with new applications, beyond water pumping in mines), and it increases the genericity of machine tools (now working beyond the limits of the usual energy source, hydraulic energy provided by rivers). (2) simpler illustrative case: the “swiss army knife” relies on the technique that enables to relate the different “tools” of the swiss knife—the articulation technique that enables to combine, for instance, a knife and a bottle opener.

2. Hence, a second, complementary perspective on technique: several authors, particularly in Engineering design, have underlined that for one given set of functions, there are different ways—different combinations—to address it, and these different combinations don’t have the same value. In particular, Design Structure Matrices (Ulrich and Eppinger 2008), or Modularity (Baldwin and Clark 2000), or Aximatic Design (Suh 2001) lead to distinguish between technological systems, although these systems seem equivalent from a functional point of view: systems with less interdependencies (DSM), with modularity (Baldwin’s Design Rules) or with independences (first axiom of Axioamtic Design) are “better”—they are said to be

more robust, easier to realize, enable a large variety of alternatives, etc. In this second stream of work, one has to consider the technique in the set of all existing techniques and the structure of this set.

How should one describe this structure of the set of techniques? Several authors have noticed that the logic of “combination” is too fuzzy to describe the structures of technical systems. Precisely, Bertrand Gilles explains that, more than combinations, there are *interdependences*, “that within some limits, as a very general rule, all techniques are, to various degrees, depending on one another, so that there should be some coherence between them” (p. 19) (Gilles 1986). And consequently, Bertrand Gilles defined a “technical system” as “the coherence, at different levels, of all the technical structures, of all the technological sets and ways” (p. 19). In this approach, the technique is not limited to its functional role but is also characterized by its *interdependences* with all other techniques. As a consequence, techniques “evolve” not only because of functional challenges and/or new scientific phenomena but also because of expected interdependences between techniques. In this second approach, market and science are not the only engines in the dynamics of technological system: the techniques themselves have their own, *endogenous dynamics*.

Studying the dynamics of techniques becomes studying the *evolutions in the interdependences in the set of techniques*. On the one hand, engineering design literature has often insisted on the importance to create non-pervasive techniques. Robust design recommends to design techniques that prevent propagations in complex systems [e.g. the use of DSM tools to control and limit interactions, (Ulrich and Eppinger 2008)]. The first axiom of Suh’s axiomatic design prescribes to design systems with minimal interdependencies (Suh 1990). Complementarily, on the other hand, Engineering design literature has also studied how, in certain cases, the creation of specific interdependences have a *systemic impact*: enabling “modularity” or “diagonal matrices” or providing a new “common core” (Gawer 2009; Kokshagina et al. 2013a), some techniques open access to a large variety of configurations—they enable a logic of platform, of modularity, etc. This systemic impact can be characterized by criteria like generativity and genericity: according to some authors, one should analyse the *generativity* of the design in a technical system, i.e. the new design paths that are opened by the new technique (Le Masson and Weil 2013; Hatchuel et al. 2011); other authors have underlined that one single technique can immediately enable a large set of products following a pure

“combinative” logic, i.e. with only very limited efforts, and hence the authors insist on the *genericity* of this technique (Kokshagina et al. 2013a; Bresnahan and Trajtenberg 1995). For instance, the rotary steam engine is generic because it suddenly enables new combinations of (existing) steam engines with every kinds of (existing) machines; it is also generative because it will further enable the design of completely new mobility systems like locomotives. These techniques are called pervasive because they propagate and perturb the set of existing techniques.

This quick overview of works on the dynamics of technologies helps us frame the research:

1. Phenomena: we already have a rich phenomenological knowledge base—we can distinguish between the design of pervasive and non-pervasive techniques, and we want to be able to account for the design of both types and for the repeated designs to describe the dynamics of technical systems.
2. Modelling techniques: understanding the design of technique requires to adopt *a model of technique* in which a technique is characterized by the interdependences that this technique has with the set of techniques.
3. A research issue: the dynamics of techniques is related to the evolution in the interdependences. Hence, very generally, our issue:
 - Q_0 : *To understand the “endogenous” dynamics of technology, we need to model strategies for the design of techniques with systemic impact (Q_0).*
4. More precisely, a first research question:
 - Q_1 : *Can we find a model that accounts rigorously for “systemic impact” on a set of interdependent techniques, and in particular for the variety of forms, including the design of non-pervasive and pervasive techniques.*
5. And a second research question:
 - Q_2 : *How does such a model predict the emergence and possibility of strategies for the design of techniques with systemic impact (pervasive techniques)? In particular, can this model at least enable to quantify such notions as “generativity”, “genericity”, “independence level” and how do these quantifiers help characterize different*

dynamics of techniques and different strategies of repeated design?

2.2 Design theory and knowledge structure

Design theories account largely for the impact of certain knowledge structures on the design of new artefacts. Historical analysis of Design Theories studied the effect of knowledge structure on the generative power in different cases (parametric design, systematic design) (Le Masson and Weil 2013). General Design Theory (Yoshikawa 1981; Reich 1995) shows how a knowledge structure with an Hausdorff measure warrants the design of any functional combination; Coupled Design Process (Braha and Reich 2003) shows that more generally, the set of functional combinations that can be reached depends on the “closure” of technologies, i.e. their neighbourhood of alternative technologies; in Axiomatic Design, when the DPs and FRs meet the first axiom, then larger ranges of functional requirements can be easily reached; in Infused Design (Shai and Reich 2004a, b), design capacity increases with the rules in one domain and the laws linking different domains; in C-K theory, the generative power comes from “holes” in the knowledge structure (Hatchuel et al. 2013); in Forcing, the knowledge base has to meet the “splitting condition” to enable the design of a set that is different from all already known sets (Hatchuel et al. 2013; Le Masson et al. 2016).

This review shows that *given a structure of dependences and independences in knowledge, Design Theories help to predict the impact of this structure on design capacities.* This impact can be characterized by the increase (or decrease!) in the capacity to design further original design (*generativity*) and in the capacity to obtain a large set of artefacts by combination of the newly designed technique with all previously known techniques (*genericity*). *However, they don’t offer yet a computational and quantitative approach of generativity, genericity, dependence and independence.*

Now for our research topic, the question is more precisely: how do design theories model the design of a new “technology” characterized by its systemic impact? Or: how to design a specific “knowledge structure”? Actually design theories tend to favour the design of artefacts, given a certain knowledge structure; but they barely address the issue of designing a specific knowledge structure. Some insights are given in (Kokshagina et al. 2013a), based on a study of algebraic extensions with C-K theory. This study was actually made possible by one key property of contemporary design theories: they don’t depend on specific objects or domains—knowledge is a “free parameter”.

Hence, even if design theory did not focus until now on the design of specific knowledge structures, design theory is a favourable framework to study this question. *In this paper, we will use design theory to analyse design strategies applied on specific knowledge structures.*

Note that this will imply a theoretical restriction: C-K theory actually explains that if there is a “theory of the object” in K0, then design theory will account for the *revision* of this theory of the object. If one wants to analyse the transformation of objects *inside* one stable framework, then this framework shouldn’t be changed by design, it has to be idempotent by design. This was the case with extension of algebraic structures: actually the extensions were all made into the complex field. This will also be the case in the rest of this paper: we look for a “theory of a set of techniques” that is relevant for the study of the design of a technique and its impact on a set of interdependent techniques *but* this theory has to be stable by design. Hence, in the rest of the paper we will check that our model meets this “idempotency” condition.

2.3 Knowledge structures characterized (only) by independences: why relying on matroids

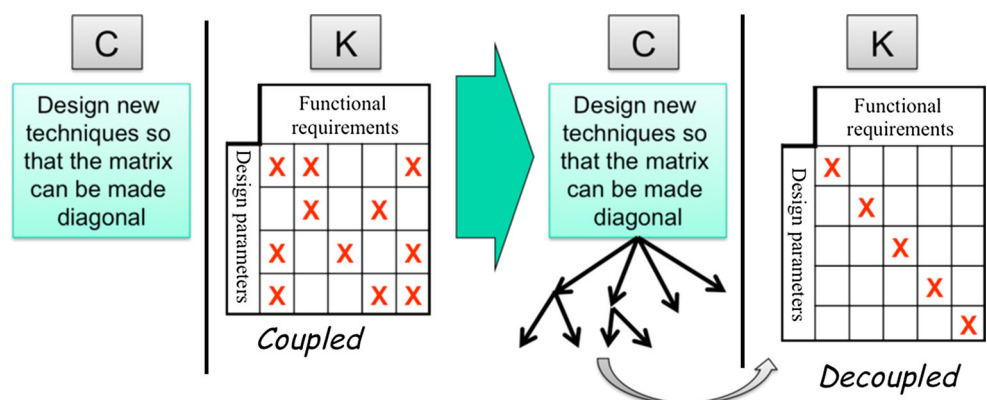
What is the relevant knowledge structure to study design strategies with systemic impact? As seen above, we aim to *characterize this knowledge structure by dependences and interdependences*, and we expect that the knowledge structure is such that generativity and genericity can be computed.

Let’s take an example: in an axiomatic design problem (a non-diagonal system, i.e. a system with linear interdependences), we can use a design theory (like C-K design theory) to design a system that meets the first Axiomatic Design axiom, hence is now diagonal (see Fig. 1 below)—in C-K, we take as knowledge base the initial non-diagonal system with its interdependences; the design process transforms the initial knowledge structure into a new knowledge structure with new independences. In this case,

the knowledge base is characterized by specific interdependences in linear algebra. Let’s take another example: we could take as knowledge base a functional graph (with graphical relations between functions) and design a graph with less interdependences. We use design theory to change the dependence relations in a graph or between vectors. Yet, in both cases, we are actually only interested by the *evolution of dependence relations*—be they based on linear algebra or graphs. Hence our analysis would be more general if applied to a knowledge structure that is only characterized by the interdependences between the elements—whatever the deep nature of the relation (graph or linear algebra). By chance mathematicians have already studied such a strange object in great detail: they call it *matroid*.

Matroid structures were introduced by Whitney, in the 1930s (Whitney 1935), to capture abstractly the essence of (linear) dependence. Whitney explains his project as follows: “let C_1, C_2, \dots, C_n be the columns of a matrix M ”. Any subset of these columns is either linearly independent or linearly dependent; the subset thus falls into two classes. These classes are not arbitrary; for instance, the two following theorems must hold: (a) any subset of an independent set is independent; (b) if N_p and N_{p+1} are independent sets of p and $p + 1$ columns, respectively, then N_p together with some column of N_{p+1} forms an independent set of $p + 1$ columns [...]. Let us call a system obeying (a) and (b) a ‘*matroid*’”. (p. 509) (Whitney 1935). Hence, the description with matroids will be very general (and quite poor—as Whitney: “The fundamental question of completely characterizing systems which represent matrices is left unsolved”) but it has the great advantage of only characterizing the relationship between elements in two modes: independence and dependence. And this remains valid for a large set of objects—Whitney: “In place of a matrix we may equally well consider points or vectors in a Euclidean space, or polynomials, etc.” and more recently: graphs, matrices, groups, algebraic extensions, etc. (for a pedagogical introduction to matroid, see (Neel and

Fig. 1 Designing a diagonalized matrix—*left-hand side* before design; *right-hand side* after design, the knowledge structure has changed



Neudauer 2009); the reader can also refer to the table presenting the definition of the main matroid notions used in this paper, in “Appendix”). In particular, it means that our two examples above (with vectors in Suh axiomatic or with functional graphs) could be addressed with the same formalism.

In matroid, *independence is not based on a specific type of relation*. We don’t need to specify the relation: it is enough to know that there is such a relation as soon as, for a finite set E , there is a collection I of subsets of E such that I satisfies the following properties: (1) I is non-empty; (2) every subset of every member of I is also in I (I is *hereditary*); (3) if X and Y are in I and $|X| = |Y| + 1$ (the operator $|\dots|$ 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). The elements of I are called the independent sets of $M(E)$, a matroid on E .

Note that the third axiom is critical: if one keeps only the two first axioms, one has a very general theory of independence—but it can’t account for usual mathematical independences in graphs or vector spaces. The third axiom (axiom of independence augmentation) puts a strong constraint on the independence structures but helps account for classical specific models of independences.

Some comments:

- Independence or dependence in matroid theory is *not* a property of an element but it is a property of a subset of elements.
- This axiomatic seems restrictive in the sense that not all sets of elements follow these axioms. Actually this is the condition to speak of independence with rigor—more precisely with the same rigor as one speaks of independence in vector space or independence in graph (or in several other mathematical formalisms). And more generally, the matroid axioms are actually valid for a large range of relations.
- In this paper, we will often illustrate the results with *graphic matroids*, i.e. a matroid that can be represented by a graph. Not all matroids are graphic. Still the results are actually valid more generally for every form of matroid. For example, we will apply our results to uniform matroid that are not graphic in general.

Hence this paper aims at understanding the “*endogenous*” dynamics of technologies by applying design theory on a knowledge structure modelled with *matroid*. We first model elementary operations (part 3) and show that these elementary operations correspond to basic phenomena in the design of techniques (and still follow the “idempotency” condition mentioned in 2.2: the design doesn’t modify the matroid model)—then we propose quantifiers to model the impact of the design of techniques on a set of

interdependent techniques—this leads to analyse the variety of strategies for the design of techniques that impact a set of interdependent techniques and this leads to propose new laws for the dynamics of a set of interdependent techniques.

3 A C-K model with matroids

3.1 Notions of techniques, systems and independences in a matroid framework


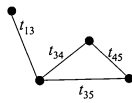
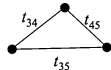
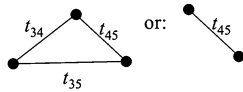
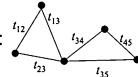
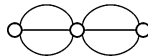
To build a C-K model with matroid, we introduce the matroid of known techniques as the K-space in C-K. Assume E , a list of known elementary techniques: T_1, \dots, T_n . We select a subset of this set of elementary techniques. If they build a working system,¹ we will say the subset is *dependent*; if not, it is *independent*. A working system can be minimal when all subsets are independent—it is impossible to remove one element without transforming the dependent set into an independent one. Such minimal dependent set is called a circuit in matroid theory (and it is called a cycle in graphic matroid). Hence a working system contains at least a circuit. And we call a minimal working system a working system that is a circuit.

Note that the notion of “*dependent*” relates also to a very basic notion for a minimal working system: we say that a (minimal) working system is a “dependent” set of techniques because (for the techniques that belong to one minimal circuit contained into the working system) if one technique is out of order, i.e. it is removed from the working system, then the system doesn’t work anymore, it is no more a working system, and it becomes an independent set. One can say that *the existence of the minimal working system “depends” on the set of the techniques that compose it*.

In general, a set of elementary techniques is *not* necessarily creating a working technical system—in matroid terms: the elements are not necessarily dependent. Still we could consider one particular case, a “lego-like” matroid of technological system, in which every set of two (or more) elementary techniques is said dependent (i.e. it creates a working “lego”)—this corresponds to a specific matroid, called the uniform matroid $U(1, n)$ (where n is the number

¹ We have here a terminology issue. We will distinguish two “systems”, inherited by two disciplines. On the one hand, Bertrand Gilles, as an historian, speaks of “technical system” to designate the set of all techniques and their relationships. This is what we model with a matroid, the matroid of all known techniques. On the other hand, the authors of engineering systems speak of a “system” to designate a working assembly of techniques—this is a subset of the set of techniques, this set having the “working” property. This is what we call a “working system” in the matroid model. This is actually a dependent set of the matroid. The “smallest” dependent set are circuits (also called cycle in a graphic matroid) (see Table 1).

Table 1 Notions in the dynamics of techniques and notions in matroid theory

Dynamics of techniques	Matroid theory	Illustration with graphic matroid
Elementary technique	An element in a matroid	
Working system: a system made of compatible techniques (techniques that work together)	A dependent set, contains at least a circuit (a cycle)	Extracted from M below 
A minimal working system: each technique is indispensable for the working system to work	A circuit (a cycle in graph matroid)	Extracted from M below 
A family of techniques: a subset of techniques such that no “external” technique is compatible with the techniques in the family	A flat, extracted from a matroid	Extracted from M below 
The structure of techniques	The matroid (of techniques)	Matroid M: 
The structure of minimal working systems	The dual of the matroid of techniques	

of elements in E). Hence, a first result: *a simple “lego-like” combinatorial model corresponds to the uniform matroid.*

To give a “visual” example: suppose that the matroid of techniques is graphic. It means that it can be represented by a graph G where each elementary technique is an edge t_i [$E = E(G)$, i.e. the set of edges of the graph] and the circuits of the matroid are the cycles of the graph. Following the definition above, a *working system contains a circuit in the graph* (a path of elementary techniques that is connected and all vertices are of degree 2, i.e. the circuit goes only once through each vertex). This means a set of elementary techniques will be considered as a “working technical system” if and only if it contains a *circuit* that links some of them. It is a minimal working system if it is a circuit.

It is important to underline that this model fits perfectly with our requirements on a model of techniques (see Sect. 2.1): in a graphic matroid model, an elementary technique, i.e. an edge, is characterized by its relations with the other techniques—i.e. whether it builds dependent or independent sets with other edges, i.e. whether it can be included in a working system with other techniques.

For instance, the graph G below can be interpreted as a synthesis of the technological know-how of a designer. The designer knows the working system $\{t_{12}, t_{13}, t_{23}\}$; he doesn’t know any minimal working system involving $\{t_{34}, t_{45}\}$. A matroid can be associated to this graph of a designer’s knowledge, the matroid defined by the cycles of the graphs. In this, matroid $\{t_{12}; t_{13}\}$ is independent, whereas $\{t_{12}, t_{13}, t_{23}\}$ or $\{t_{12}, t_{13}, t_{23}, t_{45}\}$ is dependent. Note that $\{t_{12}, t_{45}\}$ is also independent.

We model the dynamics of technologies in matroid basic notions: a technique, a working system, a family of techniques, the structure of all techniques, and the structure of all working systems (see below).

Note that matroid theory provides immediately one quantifier: a matroid has a certain rank, which actually corresponds to the size of the largest independent set. In a graph G , we have the rank function $r(M(G)) = |V(G)| - 1$. Here $V(G)$ is the number of vertices of the graph G . $r(G) = 4$ in the example above. The rank corresponds to the highest number of techniques in an independent set: at most the graph G above enables to gather (in a independent set) four edges, hence four “independent elementary techniques”.

We have now a model of a set of techniques as a matroid. This raises some comments regarding the modelling logic: to what extent does it capture the reality of techniques?

1. Like always with a model, it doesn’t account for every aspects of techniques—it more precisely aims at accounting for one critical aspect we want to address in this paper: interdependencies.
2. One can at least note that it is compatible with other models of techniques we find in the literature: for instance in Suh axiomatic design techniques are design parameters that address multiple functions—hence they are vectors and they can be independent as vectors are independent in a vector space. One can describe this logic of interdependencies of Suh axiomatic with a matroid model. As will be explained in more detail later, a set of n DPs that follow the first

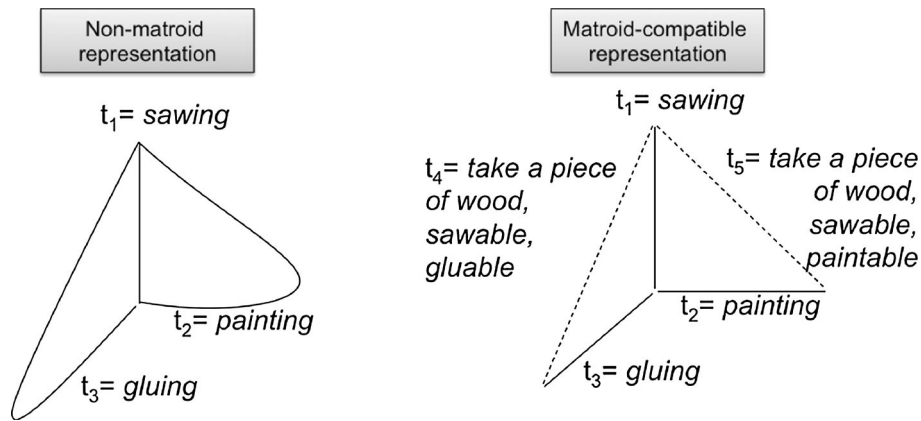


Fig. 2 Matroid perspective to support the analysis of technical object. *Left* one takes into account three techniques t_1, t_2, t_3 . t_1 and t_2 are used both in a working system, t_1 and t_3 are used in a working system, but t_1 and t_3 are independent: this doesn't fit with a matroid model—and actually cannot be drawn with a graph. *Right* the matroid

model leads to identify t_4 and t_5 such that the working systems are preserved and the dependent relationship between techniques is preserved: t_1 and t_2 are in a working system, t_1 and t_3 are in a working system, t_2 and t_3 are independent

axiom to meet n FRs can be represented by the $U(n, n + 1)$ matroid (see below, paragraph 4.1.1 and 4.3.2).

- Every set of techniques with relationship between techniques is *not* necessarily following a matroid model. We thank one reviewer for the following simple counter example: suppose a set of techniques $\{t_1 = \text{sawing}, t_2 = \text{painting}, t_3 = \text{gluing}\}$ where $\{t_1, t_2\}$ and $\{t_1, t_3\}$ are said “dependent” (by sawing and painting and by sawing and gluing one obtains two types of wooden toys) and $\{t_1\}$, $\{t_2\}$, $\{t_3\}$ and $\{t_1, t_2\}$ are said “independent”. Then, this set doesn't follow the matroid axioms (by the independence augmentation condition, $\{t_1\}$ and $\{t_2, t_3\}$ are independent hence $\{t_1, t_3\}$ should be independent). Still the matroid model rather helps to reason with rigor on the notion of independence: on the example above, the matroid model leads to say that the notion of “independence” used in this example is not well-formed in the sense of matroid, and hence, there is a risk to lead to internal contradictions when one then uses the matroid model.
- Reciprocally, one can also try to model a set of techniques in such a way that it fits with a matroid model in order to be able to analyse the properties of the model (Fig. 2). In particular, in the above mentioned example, we can set the modelling issue this way: we want to propose a matroid that meets the following requirements: the set of techniques contains at least $\{t_1 = \text{sawing}, t_2 = \text{painting}, t_3 = \text{gluing}\}$ and where $\{t_1, t_2\}$ and $\{t_1, t_3\}$ are, respectively, *included in working systems* (by sawing and painting and by sawing and gluing one obtains two types of wooden toys) and $\{t_1\}$, $\{t_2\}$, $\{t_3\}$ and $\{t_2, t_3\}$ are *independent*, this time in the sense of matroid. Is there a matroid that meets these requirements? Let's introduce $t_4 = \text{“take a piece of wood that is compatible with sawing and painting”}$ and $t_5 = \text{“take a piece of wood that is compatible with$

sawing and gluing”, and such that $\{t_1, t_2\}$ is actually included in $\{t_1, t_2, t_4\}$ that corresponds to the working system “wooden toy by sawing and painting”, hence is dependent and $\{t_1, t_3\}$ is included in $\{t_1, t_3, t_5\}$ that corresponds to “wooden toy by sawing and gluing” and is also dependent (we have the same wooden toys). $\{t_2, t_3\}$ is still independent (see illustration below). This structure follows the matroid axioms. And it becomes possible to make rigorous deduction on the interdependence relations in this structure, without a risk of formal contradiction. In particular, here we say that $\{t_1, t_3, t_4, t_5\}$ is dependent which means that “ $t_4 = \text{take a piece of wood that is compatible with sawing and gluing and } t_5 = \text{take a piece of wood that is compatible with sawing and painting, and } t_2 = \text{paint and } t_3 = \text{glue”}$ can also lead to another wooden toy.

Hence, the “ordinary” language applied to techniques can be misleading when it comes to “independence”—and a matroid model of independence can also help build models of techniques in which the notion of independence is controlled and corresponds to matroid theory.

- More generally: can every set of techniques be modelled to follow the matroid axioms? We won't address this question here—this is not self-evident. In the rest of the paper, we will make the assumption that the matroid axioms apply. The results are obtained by admitting this assumption.

3.2 Designing a new matroid: a C-K design process on matroids

How does C-K apply to a knowledge base made of a matroid? We have to see how each classical notion of C-K is applied in case of a matroidal knowledge base.

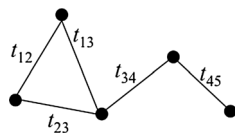


Fig. 3 Graph G

In C-K theory, K is the space of propositions that have a logical status. The logic in matroids is based on dependency: the proposition “there is a minimal working system including technique t_i ” is true if and only if there is a circuit C containing t_i . On Fig. 3, $\{t_{12}, t_{23}, t_{13}\}$ is independent, hence the proposition “there is a circuit that contains t_{12} in the matroid $M(G)$ ” is true; the proposition “there is a circuit that contains t_{34} in the matroid $M(G)$ ” is false.

In C-K theory, C is the space of propositions that are interpretable in K and undecidable in K. Interpretable: if K is modelled as a matroid, the proposition in C must also be expressed in matroid language, and hence, it is a proposition on some of the elementary techniques of the matroid. For instance: “there is an elementary technique (i.e. an edge) such that there is a circuit that contains t_{34} ”. The latter proposition is interpretable. It is also undecidable—it is neither true nor false in the matroid interpreted as a model of elementary techniques. One design for this new edge might be an edge t_{35} , or t_{25} . Hence, we can formulate a concept for a matroidal K-base in C-K.

What is now the design process? Matroid theory teaches us that there are only three basic operations—design by deletion-contraction, design by extension and design by coextension (Oxley 2011). We will show (Le Masson et al. 2015b) that these operations in matroid correspond to the three very specific, contrasted ways to design techniques that we described in the first part and are well known in the literature: designing one working system without designing a new technique, designing one new working system by designing one non-pervasive technique, or designing a pervasive technique (see synthesis table below).

This is one surprising result of C-K/Ma: it helps to clarify well-known archetypes in the design of techniques (Table 2).

3.2.1 Trivial cases: deletion and contraction

To begin, let’s first analyse the classical design issue that consists in designing one “working system” by using existing elementary techniques, while respecting their interdependences. We call it “trivial” case, because no new edge (no new elementary technique) is designed. It actually consists in “picking up” existing techniques to realize one new working system. In matroid terms, it consists in “extracting” one specific dependent set from the initial matroid by relying on two operations that keep the matroid structure, namely deletion and contraction.

How can one describe the design process? Deletion and contraction in matroid generalize deletion and contraction in a graph:

- Deletion in a graph consists in skipping an edge. Deletion is equivalent to decide to not use a elementary technique for a certain design. In the example below: to design the working system $\{t_{12}; t_{23}; t_{13}\}$, one deletes t_{34} and t_{45} . More generally, in a matroid, deletion can be defined as follows: given a matroid $M = (E, I)$ and an element e of M , deletion is the operation that forms the matroid $M \setminus e$ by removing e from E and keeping the independent sets of I that are included in $E - e$.
- Contraction in a graph G occurs relative to a particular edge, e . The edge e is removed and its two incident vertices, u and v , are merged into a new vertex w , where the edges incident to w each correspond to edges incident to either u or v . One writes G' the new graph and one writes $G' = G/e$. Contraction is equivalent, in linear algebra, to a projection in a space orthogonal to the contracted element. Contracting an edge in a cycle

Table 2 Design operations in the dynamics of technique and in matroid

Dynamics of techniques	Matroid theory	Illustration with graphic matroid
Designing one working system based on existing techniques	Extracting one circuit	
Cumulative design of working systems with new technique linking other techniques and minimizing propagations (non-pervasive)	Extension i.e. one dependent edge, depending on the techniques to be linked together	
Designing a pervasive technique, generic to several technical families	Coextension, i.e. one independent edge common to several connected components	

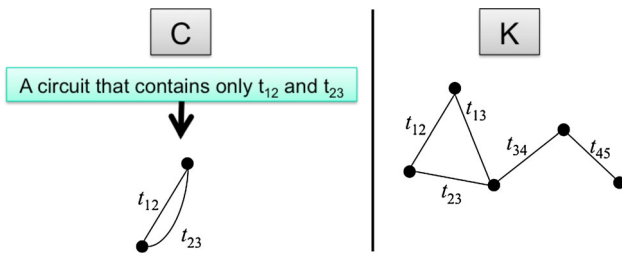


Fig. 4 Design by deletion and contraction

actually means that a working system can be “reduced” to smaller one with less elementary techniques (see Fig. 4) (e.g. a refrigerator without cooling system becomes an isothermal box). For instance the concept “a circuit that contains only $\{t_{12}; t_{23}\}$ ” is obtained by contracting t_{13} (and deleting t_{34} and t_{45}). More generally, in a matroid M , contraction actually corresponds to deletion in the dual of the matroid M . The notion of dual will be introduced later.

More interesting are cases where the concept actually leads to design a new “bigger” knowledge structure—i.e. the concept addresses the whole matroid, and not only a subpart of it and the process will “grow” the initial matroid M , in such a way that the new matroid N will “contain” the old one. For these non-trivial cases, matroid theory tells us two ways to design new edges in a matroid: extension and coextension. We analyse these two ways and their respective consequences on the structure of the matroid of techniques.

3.2.2 Extension: or the design of a non-pervasive technique

Extension is the reverse operation of a deletion: if a matroid M is obtained from a matroid N by deleting a non-empty subset of $E(N)$, then N is called an extension of M . In particular, if the subset is a simple edge $\{e\}$, then N is called single extension of M (see the example in the figure below, we add t_{35} as a new edge in $M(G)$).

What is the systemic impact of extension? To analyse the systemic impact, we need one additional notion from

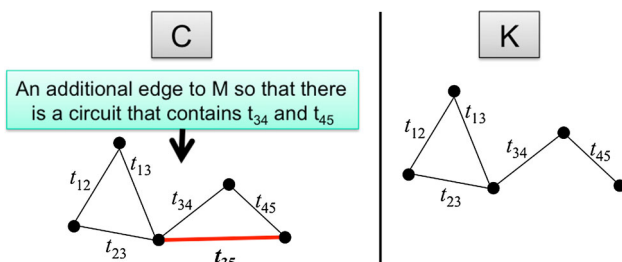


Fig. 5 Design by extension

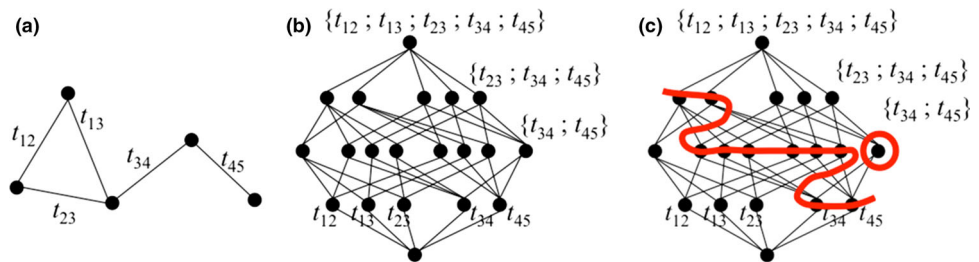
matroid theory, namely the notion of flat: a flat F is a set of elements of E such that it is impossible to add a new element of E into F without changing its rank. In the matroid M in Fig. 5, $\{t_{12}, t_{23}, t_{13}\}$ is a flat and $\{t_{12}, t_{13}\}$ is not a flat; $\{t_{34}, t_{45}\}$ is also a flat. One says that a flat is a “closed” set in a matroid. A flat can be seen as a “family” of elementary techniques that are incompatible with any other elementary technique outside the family—for instance, we can consider that the family of techniques used in automotive industry is a flat because these techniques are said incompatible with any other techniques from another industry. The set of flats forms a lattice (with the inclusion relation). This lattice of flat represents the families of technologies and their inclusion relations. For instance, one represents below (Fig. 6) a matroid and its lattice of flats.

The systemic impact can be modelled as the impact of the design of a new edge e on the structure of flats. When M is extended with a new edge e to form the matroid N , we know from matroid theory that there are three possibilities for any flat F :

- *Type 1 “e-independent flats”*: $F \cup e$ is a flat of N and $r(F \cup e) = r(F) + 1$ —it means that the new edge is independent of the flat. If t_{35} is added in G above, then $\{t_{12}, t_{23}, t_{13}\}$ is such a flat in G . These are all the flats that are not “impacted” by the design of e . More generally, these are the flats below the red line on Fig. 6c.
- *Type 2 “e-determining flats”*: $F \cup e$ is a flat and $r(F \cup e) = r(F)$ —it means that the new edge e is actually dependent on the edges of F . If t_{35} is added in G above, then $\{t_{34}, t_{45}\}$ is such a flat in G . These are the flats above the red line on Fig. 6c. Note that these flats follow a certain structure: they form the so-called modular cut, which means that they are ordered by inclusion and there is a smaller flat, which is included in all other type 2-flats (in our case: $\{t_{34}, t_{35}\}$ is included in $\{t_{12}, t_{23}, t_{34}, t_{45}\}$). This smaller flat actually determines the new edge [the demonstration would require more space, see theorem 7.2.3 in (Oxley 2011)].
- *Type 3 “e-determined flats”*: $F \cup e$ is not a flat of N —it means that some other elements should be added to $F \cup e$ to “close” it into a flat, i.e. there are “old” edges of M that are now dependent on elements of $F \cup e$. e has created new dependencies. One example is $\{t_{34}\}$ in G : $\{t_{34}, t_{35}\}$ is not a flat, since t_{45} can be added without changing the rank. These are the flats on the red line in Fig. 6c.

Hence, the impact of extension on the lattice of flat: (1) the extension “keeps” several flats unchanged (type 1 flats). (2) The effect is confined to one “smaller” flat (here $\{t_{34}, t_{35}\}$) and all the flats that contain it, and in these flats, the extension does not change the rank—hence it creates

Fig. 6 **a** A matroid, **b** its lattice of flats and **c** the effect of the extension t_{35} on these flats



new dependences (same rank with more edges), i.e. it creates new cycles, i.e. it creates new “working systems”. *Extension corresponds to the design of a technique that creates new systems inside one family (and inside all the families that contain it) and is non-pervasive for all the others. This is a non-pervasive technique that creates new working systems.*

What is the design strategy? The concept: designing a new working system. With the elements given above, it appears that extension actually corresponds to *one* new dependence inside one target family of techniques (the smaller type-2 flat). Hence an extension corresponds to a *concept*: “given (at least) two independent techniques in the matroid of techniques M , designing an edge e that is added to M to create in M a new working system that contain these techniques”.

This result helps to understand very important distinctions in the notion of “combination” when dealing with the dynamics of techniques:

1. *we can distinguish between a “new” working system (a “new combination”) and a working system that was deducible from the existing techniques (a “known combination”): extension corresponds to the design of one new working system; the new working system was not in the matroid of techniques, it was not “decidable” in the initial matroid. Otherwise, the working system was not a concept, and it was already true in the matroid.*
2. *we can distinguish between the design of a new working system (by a combination of deletion-contraction) without taking into account the interdependencies with other techniques, and the design of a new working system taking into account all the interdependencies with previously known techniques. The former case would correspond to the design of $\{t_{34}, t_{35}, t_{45}\}$, “extracted” from M (“forgetting” all the properties of dependence or independence with all the techniques); by contrast, an extension is driven by the design of one new working system but it also “controls” the impact on the whole matroid of techniques. The final result of the design is not limited to $\{t_{34}, t_{35}, t_{45}\}$ but is the matroid $M \cup t_{35}$*

(that contains in particular the working system $\{t_{34}, t_{35}, t_{45}\}$ but also all the dependences and independences).

To conclude on extension, its design and its systemic impact: *extension models the design of a “non-pervasive” technique; extension models the cumulative, non-pervasive creation of new working systems.*

We could say that a good engineering department should design by extension: based on (the matroid of) known techniques, it designs one new technique to get a new working system, it cumulates the knowledge techniques and designs the new technique for the new working system by minimizing the impact on the other techniques. By contrast, deletion-contraction corresponds to one single project that takes advantage of existing techniques but doesn’t assure backward compatibility of the new working system with initially known techniques.

Note that we better understand how an engineering department deals with combination:

- (a) It invents a new combination that did not exist before (there is a new edge)—this is different from the identification of a combination that was already known.
- (b) This combination takes into account the interdependencies with all other techniques—this is different from an opportunistic extraction of techniques without taking care of compatibilities.

3.2.3 Coextension: or the design of a technique with systemic impact

Coextension is the reverse operation of a contraction. In matroids, N is a coextension of M if there is some set T such that $M = N/T$. Since a contraction operation in matroid is analogous to a *projection* (along T) [see (Oxley 2011), chapter 3.3]. *Coextension is analogous to the reverse of a projection* [see (Oxley 2011), chapter 7.3], i.e. an *expansion*.

This operation can also be seen as an *extension of the dual* of the matroid: if N^* is an extension of M^* , then N is a

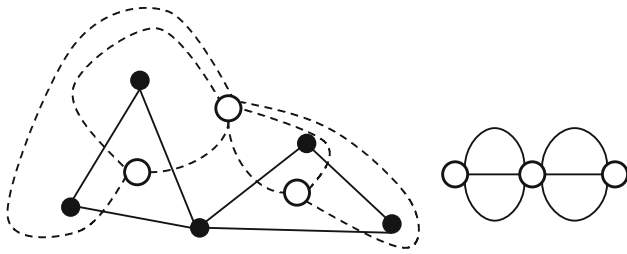


Fig. 7 Graph and its dual. On the left, black dots the initial graph G ; in white dots and dotted lines on the left the dual G^* of G . On the right, the same dual G^* without the underlying G

coextension of M . And we know quite well now what extension means in a matroid. Hence understanding coextension requires to understand what is the dual of a matroid. What is the dual M^* of a matroid M ? Formally speaking, M^* is defined over the same set of edges as M ; it can then be defined by its bases (a maximal set of independent edges—i.e. from this set one can regenerate all the other techniques that are dependent on this maximal independent set): given a basis B of the set of basis of M , a basis of M^* is the set obtained by $E(M)-B$ (this definition is acceptable because it can be proven that the set of all the “complementaries” of basis of M forms a set of basis of a matroid, that is called M^*). The dual of a matroid M can also be defined by the connected components of M . We speak here of 2-connectedness (usual connectivity in a graph is 1-connectedness): a connected component of a matroid M is a matroid N of M such that for every pair of distinct elements of N , there is a circuit containing both. In the particular case where the graph G is planar (no edges are “superimposed” in the plan), it is possible to represent the dual as in the Fig. 7. If one studies a matroid of techniques, then connected components correspond to working systems (elementary one: these cannot be obtained taking two smaller working system), hence the dual is the set of relations between (elementary) working systems.

Hence coextension consists in putting two previously independent (elementary) connected components into a bigger one, without creating a new (elementary) working system. In terms of working systems: $\{t_{12}, t_{23}, t_{13}, t_{34}, t_{45}$,

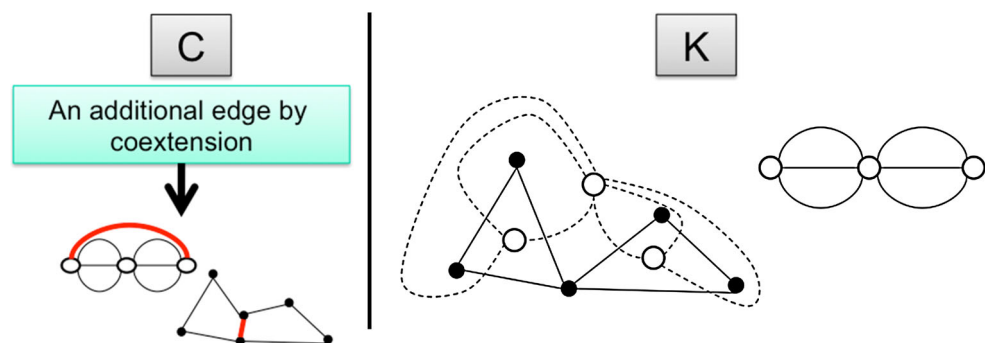
$t_{53}\}$ was a non-minimal working system; it becomes a minimal one. Designing an additional edge in a matroid by coextension can be represented by the figure below: in K , the matroid and its dual; in C , one possibility to extend the dual and to deduce the coextended matroid N (Fig. 8).

We can illustrate the coextension logic, and its difference with extension, on the practical example below. Suppose one knows techniques to make knives and techniques to make bottle openers. By extension, one can take one technique of the “family” of knives and another from the family of bottle opener and make them dependent by designing one additional technique—namely the insertion of a bottle opener in the knife handle: this is an extension. A coextension is a technique to use all the previous knives and bottle opener techniques: the articulation of the tools on one handle is the technique that enables to design swiss army knives (Fig. 9).

What is the systemic impact of coextension? Let’s analyse the impact of coextension on the matroid, contrasting it with extension (we won’t demonstrate the properties below, they are relatively classics in matroid theory [see (Oxley 2011)]—we rather insist on their consequences in the perspective of the dynamics of technologies]:

- (a) Extension preserves the rank and creates a new dependent edge in a flat—hence extension creates a new working system using the new technique. We can consider that the new working system is a “direct value” of the extension. By contrast, coextension creates a new independent edge and increases the rank. This means that the new elementary technique (the new edge) is not included in a “new” working system combining elementary techniques that couldn’t be combined before. If one considers that the direct value of design is in the new working systems that use the new technology—then coextension doesn’t create direct value! The value created by the new edge is not self-evident.
- (b) Extension is non-pervasive: it modifies only the flats that include the newly created working system. By contrast, coextension doesn’t create a new working

Fig. 8 Coextension of a graph G



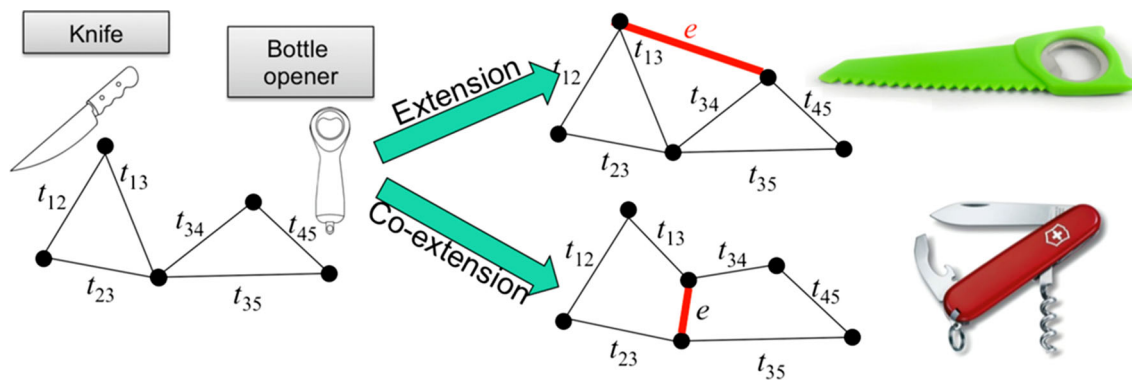


Fig. 9 Illustration of the extension and coextension on one simple case

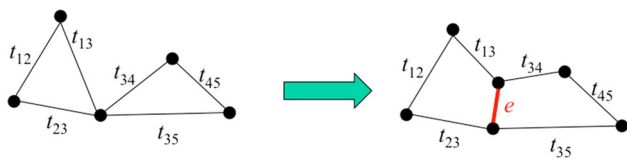


Fig. 10 Coextensions of a graph G

system with the new technique and, even worse, it disturbs “old” working systems²! In the Fig. 10, the coextension transforms a working system $\{t_{12}, t_{13}, t_{23}\}$ into an independent set (hence no more a working system) and it is necessary to add e to the system to make it work again. *The new technique is now required to make work systems that worked without it before! We could speak of value destruction.*

- (c) The new edge connects connected components that were independent before, i.e. the new technique enables “bigger” working systems by aggregating smaller working systems. The new edge integrates known working systems into a bigger one. Strangely, enough the new, “bigger” working system does *not* contain the new edge! It enables to combine into one working system two previously known, but independent, working systems. This is the critical property of coextension: *it “combines” working systems.* As such it is *pervasive*.
- (d) Working systems combination is “*modular*”. For one “old” working system (say $\{t_{12}, t_{13}, t_{23}\}$), there are now two working systems alternatives: $\{t_{12}, t_{13}, t_{23}, e\}$ and $\{t_{12}, t_{13}, t_{34}, t_{45}, t_{53}, t_{32}\}$. It means that $\{e\}$ and $\{t_{34}, t_{45}, t_{53}, t_{32}\}$ are interchangeable from the point of view of $\{t_{12}, t_{13}, t_{23}\}$. *Hence coextension creates what engineering usually calls “platform” and “modularity”.*

² We consider here co-extensions that are neither loops nor parallel edges in the dual.

- (e) Coextension increases the rank. As a consequence it opens new possibilities for extension.

What is the design strategy in coextension? The concept: designing a generic technique. With the elements given above, it appears that coextension consists in creating a relationship between (at least) two working systems to create a new working system that keeps all the properties of the aggregated systems (all previously known elementary techniques are present) and is modular. In C-K theory, the concept that corresponds to coextension is: *designing a technique that enables a working system that combines working systems that were independent until now.*

Hence, we have shown that coextension in matroid exactly corresponds to the concept of the design of a generic technique, a technique designed to be generic to several, ex ante independent, working systems.

Let’s underline two surprising properties of the generic technique:

- (a) As expected from a “generic technique”, the new technique is *pervasive*, it has a *strong systemic impact*, and it combines working systems and enables “bigger” systems. But paradoxically it is *not* visible in the new working system! (e.g. in the figure above: $\{t_{12}, t_{13}, t_{34}, t_{45}, t_{53}, t_{32}\}$). The new technique is “hidden” by the new modules and platforms. It only appears when one shifts from one module to another. As we will confirm in the illustrations below: in steam engine history as well as in semiconductor techniques history, a generic technique is not a “big”, new, breakthrough working system emerging out of nowhere—it is a discrete technique that helps combine existing working systems while changing them. It corresponds to a form of “creative destruction”.
- (b) Based on results from matroid theory, it is possible *to increase the number of independent working*

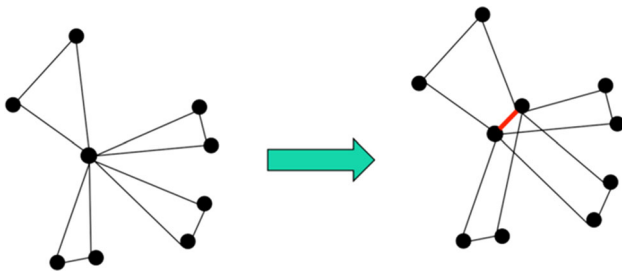


Fig. 11 Designing a technique that is generic to four independent working systems

systems combined by one new technique. Hence the *genericity can be designed.* Genericity is not necessarily the result of the random aggregation of working systems. We show below one such generic technique (see Fig. 11).

3.3 Main conclusions on elementary operation in C-K/Ma

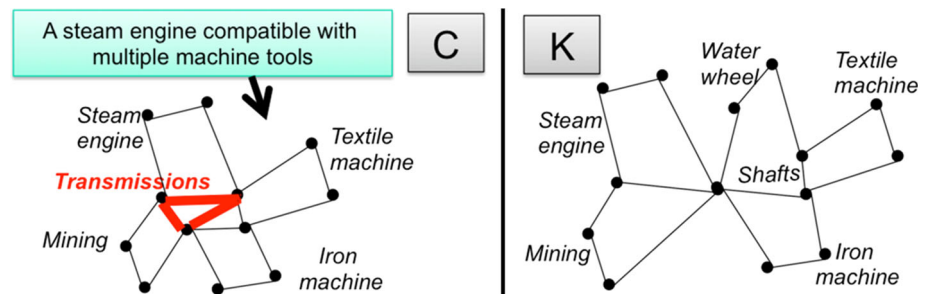
To conclude this part: we built a C-K model with matroid to study the endogenous dynamics of techniques. Note that this model meets the idempotency condition mentioned above: Matroid theory is kept stable by design in C-K/Ma. *This model enables us to considerably enrich the representation of this dynamics and the models of “combinations” in design:*

1. the model accounts for some critical distinctions in the design of techniques and technical systems.
 - (a) it accounts for the *distinction between deduction and construction.* Deduction consists in proving that one working system (circuit) exists in the matroid; construction consists in creating techniques to design one new circuit (a new working system).
 - (b) it accounts for the distinction between designing a “stand alone” working system based on existing technique (one “project”) and designing a structured set of techniques: the first one consists in extracting one working system from the matroid of known techniques to create a stand alone working system—and we can show that it is possible for all the “minors” of the matroid; the latter consists in enriching the matroid of techniques.
2. the model clarifies critical distinctions that design theory makes between combinatorics and generativity.

The model enlightens different forms of combinations, from non-generative combinatorics to generative combinatorics:

- (a) The identification of one circuit that already exists in a matroid corresponds to non-generative combinatorics
 - (b) Designing one artefact from existing techniques, by deleting some interdependencies with other techniques corresponds to a (limited) generative combinatorics (some circuits are created that where not circuits in the initial matroid of techniques—but no new edges are created).
 - (c) Extension consists in creating a new edge (new technique) to create a new working system, taking into account all interdependencies. This is a generative combinatorics: a new edge and a new working system are generated in the process. The new edge is actually dependent on the previously known one, and its design is driven by a concept of working system.
 - (d) Coextension consists in creating a new edge that increases the dimension of the whole set of techniques, i.e. create new opportunities to invent new combinations. It is a higher level of generative combinatorics.
3. Regarding phenomenology: the model accounts for two basic, very different processes in the design of techniques, the design of pervasive or non-pervasive techniques:
 - *extension consists in designing one new working system in a non-pervasive way* (minimizing the systemic impact). It can be compared to the activity of an engineering department regularly designing new working systems in a cumulative way, avoiding the propagation of changes to all its elementary techniques.
 - *coextension consists in designing a generic technique enabling the (pervasive) combination of previously independent working systems.* It can aim at combining the maximum number of working systems, and hence, it aims at maximizing its genericity. It doesn’t create “direct” value (the new technique is not involved in any new working system!), it is disturbing the “old” working system, and it recombines the working systems in a modular way.

Fig. 12 Designing steam engine as a generic technology



3.4 Historical illustration: the genericity of steam engine modelled with matroids

With the illustration below, we show how matroid models help to account for one famous historical case of designing a generic technology and how it confirms the most paradoxical properties of generic techniques modelled in matroids.

The story of the steam engine is often told this way: Watt designed a steam engine, and progressively many applications were found for it. Yet, as shown in (Dickinson 1936; Thurston 1878) and analysed in detail in (Kokshagina et al. 2012, 2013a), this story does *not* correspond to how Watt and Boulton designed a “generic” steam engine. We use C-K/Ma to enlighten some facet of this design.

In what follows we assume that the set of techniques Watt and Boulton were dealing with follows a matroid model.

The first generation of steam engines was adapted to mining, but not to other uses; hence, there was no “steam engine” in the 1770s, but only *water pumps for mining*—and Watt was first famous, in 1770s for greatly enhancing Newcomen “*fire pumps*” with a separate condensation chamber. Hence, in 1770 one can represent the matroid of techniques by the graph in K-space in Fig. 12. C-K/Ma predicts that the connected component of “fire pump” (i.e. steam engine + mining) and the connected component of machining (water wheel + textile machine + iron machining such as turning or drilling) can be connected in multiple ways but, basically, by two contrasted operations: extension or coextension. And the design of generic (pervasive) technique should be done by coextension. Hence, the hypothesis suggested by C-K/Ma: to obtain “generic steam engine”, there has been coextension in the matroid of techniques—more precisely: Ha: there should have been an initial design brief that called for a coextension; Hb: one should recognize the technique designed by coextension through several features: almost invisible in the working system, rely mainly on pre-existing techniques and take full advantages of these pre-existing techniques.

Ha: history confirms that there was a brief for coextension. Actually the story of the “*generic steam engine*” begins in the 1780s, when Boulton asked Watt to work on a new concept “a steam engine that is compatible with multiple machine tools”. And it is Watt’s new design for this concept that appeared as the “steam-engine generic technology”. Boulton’s brief was targeting a complete recomposition of the structure of techniques of his time: if a steam engine is compatible with multiple machine tools, it becomes a core component of all future machines, as all new machine tools will be redesigned to take the best advantage of the steam engine (Fig. 13).

Hb: the newly designed technique exhibits traits of technique created by coextension: the design of Watt and Boulton led to a new order of techniques based on a new technique for movement transmission (the parallelogram on the figure above). The latter appears as a “*platform*” that connects the steam engine either to mining or to workshop machine tools (in textile or iron industry, etc.). The main design effort consisted in *designing the coupling technique* (here the double acting transmission system).

This new technique has the expected traits of a technique created by coextension (see Fig. 13):

- It doesn’t emerge as a complete original technology—it actually relies on existing techniques.
- Despite its great systemic impact, the generic technique is discrete: the invention is neither in the fire engine, nor in the condensation chamber, it is only in the kinetic mechanism on top of the vertical rod.
- This is not an evolutionary process in which designers “discover” phenomena or “combine” randomly techniques: this is the intentional design of a pervasive technique (Hooge et al. 2014; Fig. 12).

Note that this example doesn’t prove that the design of generic steam engine corresponds to a C-K/Ma model. It proves that there is no incompatibility between the model and the historical traces and it shows that applying this model enables to clarify that generic technology can be

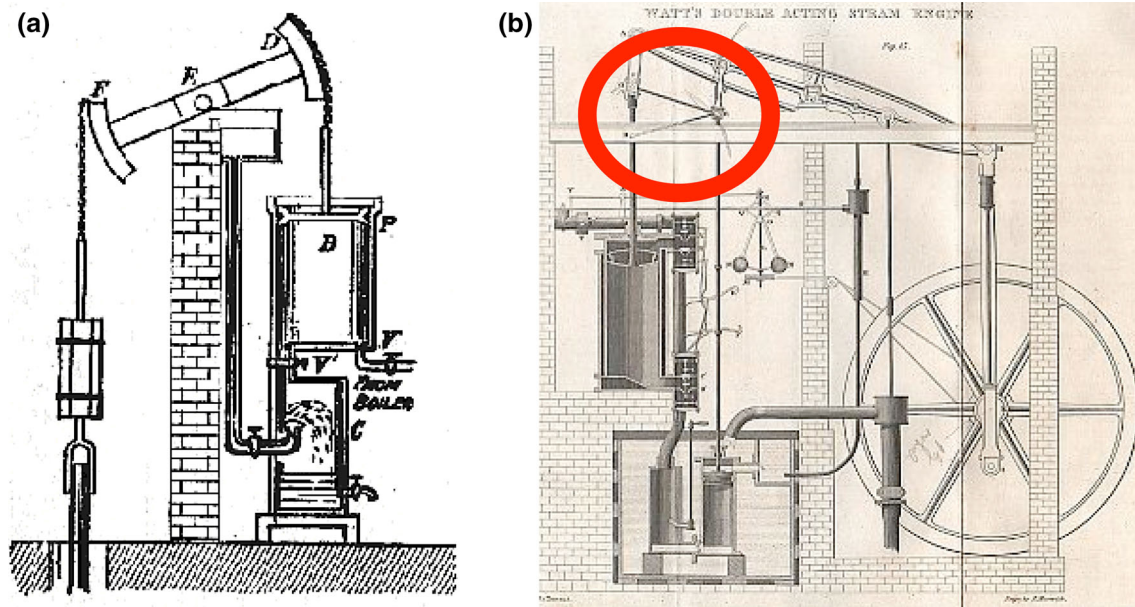


Fig. 13 **a** 1763 Watt steam engine with separate condensation chamber (not generic); **b** 1784 Watt and Boulton Double acting steam engine. The parallelogram creates genericity

intentionally designed and is not necessarily following an evolutionary processes.

4 Towards a computational model of the dynamics of technologies

With CK/Ma, we have modelled elementary design operations related to the design of a technique in a set of techniques. Based on these elementary operations (deletion/contraction, extension and coextension), we can now analyse the dynamics of repeated design in a set of techniques by build quantifiers for critical features of a set of interdependent techniques (dependence, genericity, generativity) and modelling combined and iterated elementary operations. This will provide us with elements for a computational model of the dynamics of technologies.

4.1 Characterizing features of the dynamics of technologies

The matroid approach helps to quantify and characterize complex notions like independence level, generativity and genericity.

4.1.1 Independence

With C-K/Ma, we can first *quantify independence and dependence*: the rank of the matroid in K gives the level of independence; the co-rank gives the level of dependence (the level of dependence is also the independence in the

dual, which actually corresponds to the independence between working systems). Both are linked by the equation: $r(M) + r(M^*) = |M|$ where $|M|$ is the number of edges, i.e. the number of elementary techniques.

Let's underline two special cases:

- “pure combination”: In a Lego-like matroid ($U(1,n)$), the rank is one—all techniques are dependent on the others; it means that there is no “holes”: no two independent techniques; the dual is $U(n-1,n)$, i.e. $n-1$ elementary working systems are necessary to “deduce” the last n th one. the systems are independent from each other.
- “perfectly decoupled engineering system”: Let's consider now the n techniques in a working system that follows the first axiom of Suh: the n techniques form a circuit.³ Now the first axiom applies “inside the working system”, and any subset of $n-1$ techniques follow the first axiom. Hence the set of techniques enabling a Suh working system might be represented by a $U(n-1, n)$ matroid.

³ Suh model can be directly translated in a matroid model, since it is based on vectors:

-In a DP-FR Suh matrix, one considers the DP-lines as vectors (of the FR-vector space); according to the first axiom, these vectors are independent.

-moreover, one should add one (implicit) DP: the system works. Hence a “validation” DP is “a (set of) instruments to test that the system works”. This DP has a value on each FR: it validates each FR. In a system that follows the first axiom, this last DP depends on ALL above mentioned DPs.

Hence the $U(n, n+1)$ matroid (where n is the number of FR).

Now we also want to quantify the effect of independence level on generativity and genericity.

4.1.2 Generativity

Generativity is a critical feature of any design theory (Hatchuel et al. 2011). It is usually hardly quantified. In C-K/Ma, it is possible to give a quantified evaluation of the generativity associated to a matroid M and it depends on M independence level (i.e. M rank). Generativity can be seen as a quantification of the number of new edges (techniques) that can be created on a given matroid. We know now that they are two fundamental processes of single-edge creation—hence we can *quantify the generativity by the number of possibilities of single-edge extension $g_{ext}(M)$ and the number of possibilities of single-edge coextension $g_{co-ext}(M)$* (see figure below). Note that these formulas are based on graphic matroids. Extensions (and coextensions) don't include the creation of parallel edges and loops in the matroid M (and in the dual of M).

- (a) Generativity by (single-edge) extension: it is possible to evaluate the number of single-edge extensions that can be done on a given matroid of rank r ($r(M(G)) = |V(G)| - 1$) and for $n = |V(G)|$ vertices. The maximal number of edges is $n.(n - 1)/2$ (this is the so-called *complete* matroid, where every vertex is linked by a single edge to any other vertex). If the matroid is simple (no loop, no parallel edges), there are already $|M| = r + r^*$ edges in the matroid (where r^* is the corank, i.e. the rank of the dual of M); hence, the number of possible extensions for a simple matroid M is

$$g_{ext}(M) = r.(r + 1)/2 - |M| = r(r - 1)/2 - r^*.$$

If the matroid is not simple, then there are p loops or parallel edges; hence, the simple matroid associated to M has $|M| - p$ edges and the number of possible extensions for a matroid M with p loops and parallel edges becomes

$$g_{ext}(M) = r.(r + 1)/2 - (|M| - p) = r(r - 1)/2 - r^* + p.$$

This is one first measure of the number of “extensions” (i.e. cumulative, non-pervasive design of a new working system) with a given set of techniques. *This could be assimilated to the growth potential of an engineering department knowing the matroid of techniques M .*

- 2. Generativity by (single-edge) coextension: on the other hand, it is also possible to evaluate the maximal number of coextensions that can be made from a given graph of rank r , with $|M|$ edges. We have $g_{co-ext}(M) = r^*. (r^* - 1)/2 - r + p^*$ (where

p^* is the number of loops or parallel edges in the dual of M). *We estimate here the quantity of “generic” techniques that can be proposed on a set of given techniques.* Note that there is no reason to think that it is negligible.

4.1.3 Genericity

With genericity we mean the number of applications derived from one design (Kokshagina et al. 2013a). *Genericity characterizes the systemic impact of one particular new technique.* In case of matroid, genericity can be modelled as the set of new circuits resulting from the design of a new edge e . We can characterize two contrasted forms of genericity (see Fig. 14):

- (a) We count the *new circuits created* by the design of the new edge e . In extension, the new edge creates many new circuits, in the target flat and in all the flats containing it. In coextension, we need also to count all the circuits created by the design of e and not involving e directly. This genericity results from the new combinations between connected components of elementary techniques. Since a circuit is a working system and a working system can be considered as marketable, *this genericity can be assimilated to the “direct value” created by the new technique.*
- (b) We can also count the new circuits that can *now* be created (after the design of e), with an additional effort—an extension effort or a coextension effort. In this case, this is the *“indirect value” or the “dynamic value”*. We call M the initial matroid and N the matroid created by e added to M . We will use the generativity quantification constructed above. Note that in extension and in coextension the number of loops and parallel edges is unchanged, i.e. $p(N) = p(M)$ and $p^*(N) = p^*(M)$. For the edge e we compute extension-genericity: $gen_{ext}(e) = g_{ext}(N) - g_{ext}(M)$ (and we can conversely compute: $gen_{co-ext}(e) = g_{co-ext}(N) - g_{co-ext}(M)$). We distinguish two cases:

- If e results from an extension, then we know that $r(N) = r(M)$ and $r^*(N) = r^*(M) + 1$. Hence the extension generativity has decreased by -1 with the design of e by extension $gen_{ext}(e_{ext}) = -1$. (Similarly $gen_{co-ext}(e_{co-ext}) = -1$). It just means that the number of possible extensions decreases (by -1) after each extension.
- If e results from a coextension, then we know that $r(N) = r(M) + 1$ and $r^*(N) = r^*(M)$. Hence,

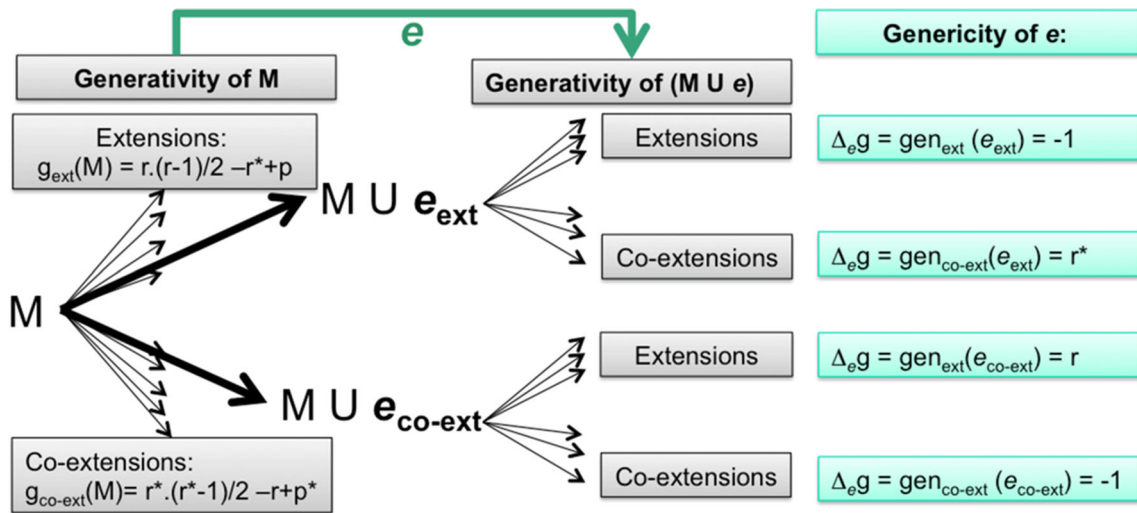


Fig. 14 Generativity of a matroid, genericity of a new edge added to this matroid

$$gen_{ext}(e_{co-ext}) = (r + 1)r/2 - r^* + p - [(r - 1)/2 - r^* + p],$$

i.e. $gen_{ext}(e_{co-ext}) = r$ (conversely: $gen_{co-ext}(e_{ext}) = r^*$). The extension-genericity of e created by coextension is approximately r . Conversely, the coextension genericity of e created by extension is approximately r^* . It means that the number of possible extensions increases (by $+r$) after each coextension.

4.1.4 Combining multiple elementary operations

Now that we have elementary operations (deletion-contraction, extension and coextension) and quantifiers for elementary operations, we can model combinations of elementary operations. We give an illustration in Fig. 15.

4.2 Illustration: designing generic technologies in the semiconductor industry

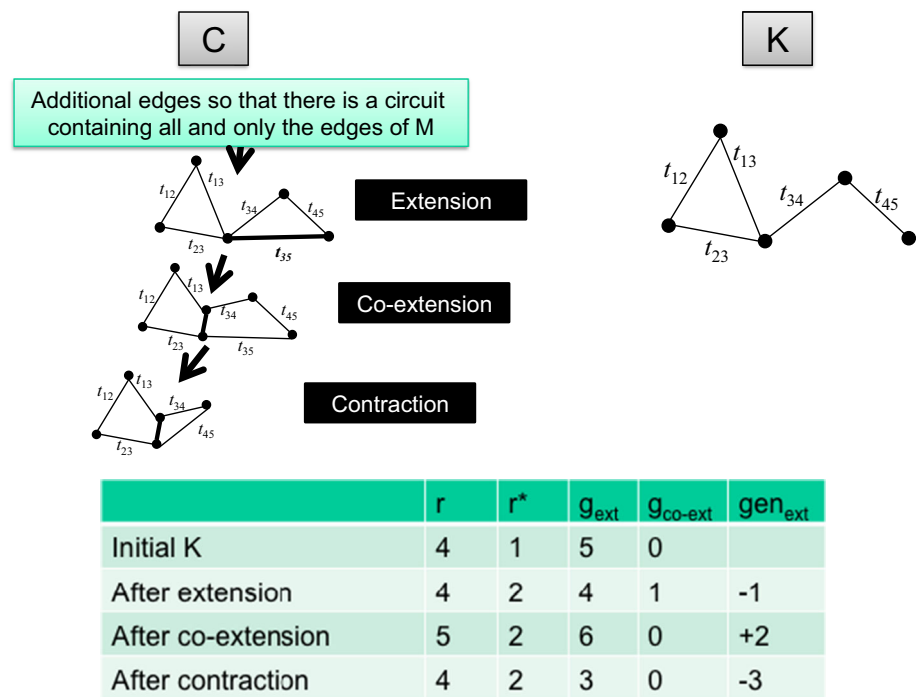
We use these quantifiers to analyse the design of generic technique in semiconductor industry. It is presented in more detail in (Kokshagina et al. 2013b). It can be schematically described as follows: there are semiconductors systems that integrate radio signals (radar, wifi, etc.) and computing power, but these systems are poorly integrated and neither use high level radio frequency sensors nor powerful computing power. The concept consists in combining three, initially unrelated, technological families:

- Family 1: the computing elementary techniques (elementary techniques for the so-called CMOS transistor).
- Family 2: the Radio-Frequency sensors, able to receive and digitalize radio frequency signals, these RF system being based on bipolar technologies.
- Family 3: the so-called “back-end” system, the elementary techniques basically in charge of routing and processing signals.

The concept is: “a matroid of elementary techniques that combine the three technological families”. By combination, one expects a system that enables circuits using elementary techniques in the three technological sub-systems (computing, RF and back-end) and circuits using elementary techniques in every pair of the three technological sub-systems (RF and computing; RF and back-end, computing and back-end). The notion of “combination” is usually quite fuzzy and hides the design issues. Applying the matroid model enables to clarify several contrasted alternatives with different “values”. Relying on extensions and coextensions, one can propose four different solutions (see below) and evaluate their potential in term of generativity and hence compare the generativity created by each design alternative.

- The “pure extension” appears as an effort to design “micro-combinations” of elementary techniques; it brings multiple working system—hence a direct value—and still none of these systems enables to combine all previously known elementary techniques. The generativity is negative: it means that the new design has decreased the generativity potential of the matroid.
- The “pure coextension” is a generic technology that creates a working system that uses all previously

Fig. 15 Design process combining elementary matroid design operations—evolution of the quantifiers of the set of techniques



known elementary techniques; moreover, it is modular (pairwise combinations are also possible). Hence there is a direct value here also—but the number of newly created working systems in coextension is rather lower than in “pure extension” case. The genericity is positive: the new design has (strongly) increased the generativity potential of the matroid.

- Hybrid case 1 is also an interesting design strategy: it exhibits a new connected sub-component that connects each of the modular ones. We see here a “platform”. Genericity is relatively high.
- Hybrid case 2 enables to get the expected performance (connections between the connected components), but it does so *by designing only two edges*, whereas all the other solutions design three edges. This is a particularly efficient “generic” technology.

The design strategy adopted at STMicroelectronics to develop the new technique actually follows that latter process: mixing sub-technologies of CMOS and bipolar into a new connected component called bi-CMOS (hence an extension) and the redesign of the back-end to connect it to the new bi-CMOS (Fig. 16).

This example shows how the C-K/Ma enriches our understanding of the design of techniques with systemic impact, supported also with quantitative criteria on generativity of the new matroids and genericity of the designed technique.

More generally, one can see on this example how computational models could be developed to simulate the possible combinations of extensions and coextensions to

modify a set of interdependent techniques—this is, however, out of the scope of this paper.

4.3 Characterizing the dynamics of techniques

With C-K/Ma, one can at least characterize specific laws of the dynamics of techniques. Note that until now, all results are obtained for any matroid (even if we illustrated mainly with graphic matroids). In what follows, we will also use other specific type of matroid, namely uniform matroid.

4.3.1 Result 1: Lego-like structure of techniques are locked and evolve only through exogenous dynamics

Let’s analyse the “intuitive” model of “combination of techniques” (used for instance in Arthur 2009). In this model, each elementary technique can be combined with another one to create a working system—hence we called it the “lego-like” structure.

Depending on the type of relation we analyse, there are two types of analysis possible: either one considers the graphic matroid G_n associated to Lego-like structure (this is a dipole with n edges). In that case, we have (self-evidently): $g_{\text{ext}}(G_n) = 0$: extension is impossible in G_n . This system is “locked” for extension. The pure combination approach prevents the design of new working systems! (actually because all working systems obtained by combinations are considered as known in the lego-like structure—i.e. any working system, i.e. any circuit linking

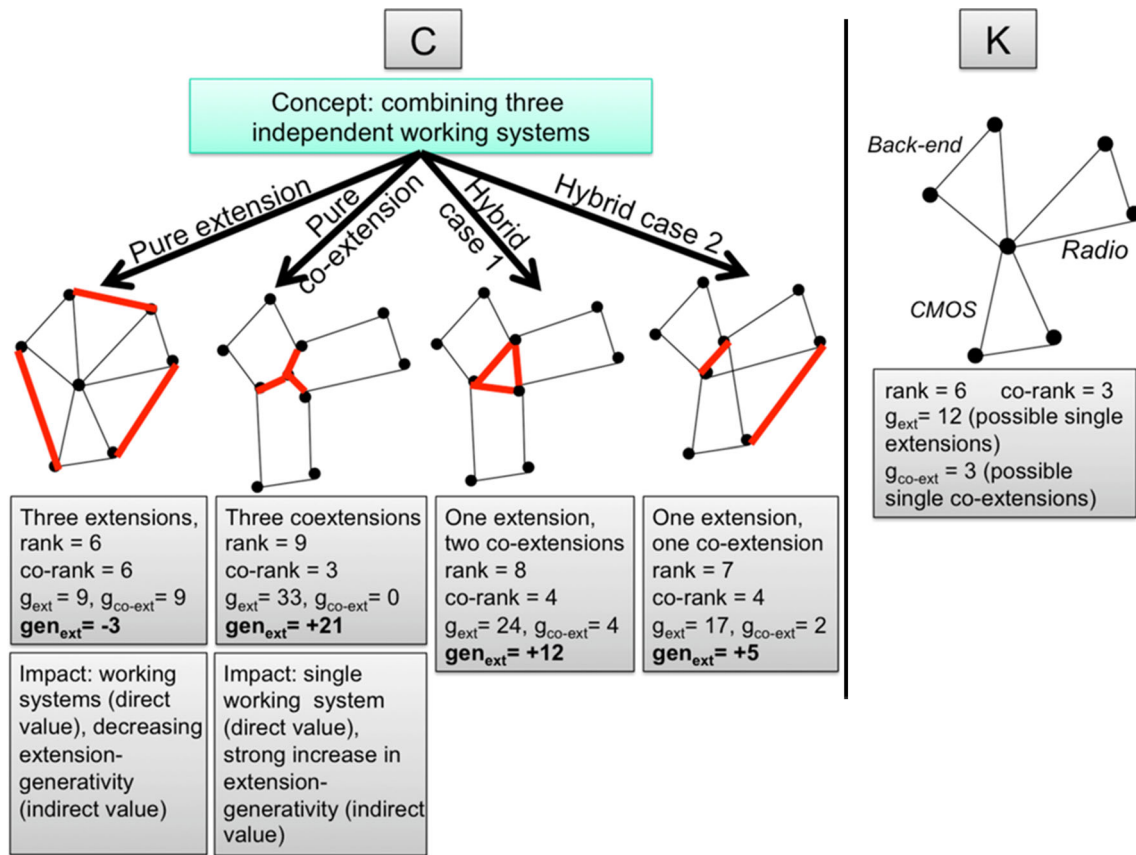


Fig. 16 Combining technologies—the case of B9MW in semiconductors

techniques from G_n can be obtained by deletion-contraction).

Still we can compute:

$$g_{co-ext}(G_n) = (n - 1) \cdot (n - 2) / 2 - 1.$$

Hence $U(1, n)$ is very generative in *coextension*. However, these coextensions are hardly visible in “lego-like” approaches *because the co-extended matroid is no more Lego-like* (the G_n family is not stable by coextension).

Another analysis is based on the fact that the lego-like structure can be seen as the matroid $U(1, n)$. One can now examine the dynamics of this specific family of techniques by considering the $U(n, p)$ family. Note that it is an interesting case for the matroid model since the $U(n, p)$ matroid is, generally speaking, not graphic so the study of $U(p, n)$ types shows how the matroid model applies beyond graphic matroids. To help the reader: the $U(p, n)$ family is defined over a set of n elements and a subset of elements is independent if and only if it contains at most p elements. A subset is a circuit if it has exactly $p + 1$ elements. It can be represented as the matroid of linearly independent subsets of n vectors in general position (i.e. “general case”, no special, coincidental cases) in an $(p + 1)$ -dimensional real vector space.

As known from matroid theory (and this is quite self-evident), the one-element extension of $U(1, n + 1)$ is $U(1, n + 1)$. It consists in adding one building block that is combinable with all previously known building blocks. A coextension is an extension in the dual. And the dual of $U(p, n)$ is $U(n - p, n)$. Hence the one-element co-extended matroid is $U(p + 1, n + 1)$. The co-extended Lego-like structure is hence $U(2, n + 1)$. It means that the Lego-like structure seen as a matroid $U(1, n)$ is not stable by coextension. Note that $U(2, n + 1)$ is not graphic.

In both models, we see that *a pure Lego-like model of the dynamics of technologies won't be able to account for technological dynamics based on coextension. Hence, a Lego-like structure cannot account for the endogenous dynamics of technological systems.* That is why “pure combination” models tend to rely on exogenous dynamics like new science (Arthur 2009).

4.3.2 Result 2: A structure of purely independent techniques implies exogenous dynamics

In the case of an “ideal” engineering system following Suh axiom, we showed above that this system might be

represented by an n -edge cycle graph, C_n , which is also the $U(n-1, n)$ matroid.

One can interpret this “configuration of technologies” either as an evolution of the n -edge cycle graph or an evolution of $U(n-1, n)$.

The reader will easily be convinced that in the graphic representation C_n , there are many possible extensions but all of them “break” the cycle, hence the family C_n is not stable by extension. The matroid is no more following Suh axiom.

This is strictly similar with $U(n-1, n)$: by extension it becomes $U(n-1, n+1)$, which is no more of the type $U(k, k-1)$ where k is a positive integer. Note again here that $U(n-1, n+1)$ is not a graphic matroid.

This corresponds to the fact that *Suh’s first axiom has a hidden effect: it prevents the design of multi-functional techniques.*

More generally, in system engineering one tends to favour systems with independent techniques—for instance, one will tend to separate mechanical, thermic, chemical or biological processes in a complex system (like a boat or a process machine). Interdependences are rather seen as defects, e.g. chemical corrosion of mechanical parts in a boat. *Hence any “perfect” system engineering will finally prevent the emergence of a multi-functional technique!*

Hence, the two simplified representations of technical systems—pure combination or pure independence—actually correspond to very particular matroids and they are actually representations that cannot account for the endogenous dynamics of techniques.

4.3.3 A necessary condition for the continuous endogenous dynamics: the combination of extension and coextension

We have shown that *extensions or coextensions, enabled alone, lead to deadlocked systems since the genericity of extension and coextension is negative* (we might have here an explanation for the “blocked technical systems” described by the historian Bertrand Gilles (Gilles 1986).

A direct consequence of this negative genericity is that the only way to get an unlocked endogenous dynamics consists in combining extension and coextension—i.e. the combination of the design of working systems and the design of generic techniques.

This combination can lead to several types of dynamics: two special regimes can be easily modelled that fit with existing engineering domains:

- The “*extension-driven*” regime gives priority to extension (the design of working systems). In this regime, coextensions (the design of generic techniques) are as rare as possible. Over time, the matroid becomes

saturated and no extension is possible anymore. Hence, one coextension is required, it increases the rank by $+1$ (the rank becomes $r+1$) and the generativity by $+r$. Over time, the rank increases slowly: one coextension that increases the generativity by r and the rank with $+1$, then r extensions until generativity decreases to 0 and again coextension, this time with the rank $r+1$, etc. The corank r^* increases with $+1$ for each extension, hence it increases a lot. Over time r becomes relatively low compared to r^* . There are finally a hand of independent techniques in a world of independent working systems. *We can recognize here the technical dynamics of automotive or aeronautic industry in the mechatronics era.*

- Conversely, the “*coextension-driven*” regime favours coextensions. We have then a symmetrical situation: a hand of independent systems and many independent techniques—but among them there are “generic” techniques that re-organize around themselves the working systems. We recognize *here the technical dynamics of semiconductor industry.*

4.4 Revisiting patent data: an illustration of the C-K/Ma dynamics of techniques

There is an open-ended debate in the analysis of the dynamics of techniques and inventions: is the source of technological novelty related to new combinations or is it related to a pure origination, with few antecedents to originate the new technological pathway? What are the relative roles of combinations and originations? (for a recent contribution to this debate see: (Strumsky and Lobo 2015)). Our model provides new answers:

1. we should necessarily find both types
2. in an “extension oriented” regime (i.e. a regime where the design favours the design of working systems), we can predict a clear dominance of combination and limited generativity. Actually, in an extreme extension-driven regime (all extensions are designed before a new coextension is done), x coextensions enable approximately x^2 extensions. 1000 coextensions would enable 1000000 extensions. These hypotheses are confirmed by the most recent quantitative study of patents (Strumsky and Lobo 2015): this study identifies the category of “origination” patents (that corresponds to coextensions) and categories for combination (that correspond to extensions). It shows an overwhelming dominance of combination, and the relationship between cumulated coextensions patent and cumulated extension patents shows a power relationship, with a power between 1 and 2, meaning that patents correspond to an extension-driven regime, still not the most

extreme one (not all extensions are patented before one coextension appears!).

5 Main findings and conclusion

In this paper, we studied the design of techniques and its effect on a set of existing techniques—an effect that can be as non-pervasive as possible or as pervasive as possible. We built a model of the design of knowledge structure by combining one design theory (C-K theory) and a model of interdependences in knowledge structures (matroid theory). This C-K/Ma model brought us following findings:

1. C-K/Ma accounts for *basic phenomena in the design of pervasive and non-pervasive techniques*. In particular, C-K/Ma helps understanding the phenomenology of generic techniques such as:
 - (a) A generic technique—like the kinetic parallelogram in steam engine—does not seem to add functional value to the system (the steam engine and the machine tools were known and did not require a kinetic parallelogram to work!) but is finally in every machines (because this is the key technique to combine the two previously independent working systems);
 - (b) A generic technique couples and decouples; it creates a “modular” relations between working systems—it reorganizes technical system in a flexible way.
 - (c) A generic technique appears first as just coupling two systems without adding direct value—but it has a strong “indirect value”, as many new techniques can then be added to the newly coupled working systems.
2. C-K/Ma, when applied iteratively, helps to propose *new laws for the dynamics of techniques and helps to build strategic alternatives in the design of techniques*. In particular:
 - a. We show that, in an iterative perspective, an endogenous dynamics would necessarily rely on extension and coextension (i.e. the design of working system and the design of generic techniques) (otherwise the dynamics would stop). It seems that historical cases (like the industries of the twentieth century) tend to rely mainly on extension processes—with rare coextensions, i.e. few generic techniques—and still it is possible to describe theoretical trajectories with regular coextensions. The model confirms also quantitative

empirical works on the source of technological novelty in patents.

- b. C-K/Ma offers a guide for the design of technologies with systemic impact, based on generativity and genericity criteria. C-K/Ma uncovers a large variety of design strategies for pervasive techniques and provides criteria to evaluate how these techniques change the generativity and the genericity of the newly created technical system.

Moreover, C-K/Ma contributes to design theory since it provides some basic quantifiers and operations that could lead to a *computational* model of the process of designing techniques with systemic impact. In particular:

- (a) In C-K/Ma, one can characterize main operations (extension, coextension, etc.) and their quantified effect (rank, co-rank, generativity, genericity).
- (b) In C-K/Ma, one can clarify the complex relationship between combinatorics and generativity, from non-generative combinatorics (identify circuits that already exist in a matroid) to generative one (generativity by extension, that results from the combination of independent techniques in the matroid of techniques and generativity by coextension, that results from the combination of independent systems in the dual of the matroid of techniques).
- (c) The model also enables to understand the limits of “intuitive” models of techniques. In particular, we show that the usual “lego-like” combinative models prevent to analyse the endogenous dynamics of technical systems. And we also show that the alternative “independence-driven” model is also limited: in engineering design, the “ideal” system maximizes independences (following Suh’s first axiom) and this system prevents the design of a generic technique. This means that if one represents an engineering system with independent techniques, then it is impossible to represent the design of a generic technique in such a model.

Regarding computationally, one has, however, to remind that matroids, in general, are not an easy object for computing. As often in model and computationability, algorithms are often more efficient when implemented at the level of specific models (vector matroid, graphic matroid, etc.). Still the logic of computationability at matroid level also helps to get very general models on the computational complexity of matroid models:

- (a) on the one hand, it is proven the efficiency of algorithms that rely on the so-called “matroid oracles” or, more specifically, on “independence oracle” i.e. these algorithms take an oracle as input

(i.e. a subroutine that tests whether a subset of elements is independent), and it is proven that they work on the variety of objects that follow the matroid axioms—this is the case for the famous greedy algorithm.

- (b) on the other hand, these formalisms were useful to prove that certain matroid problems can't be solved in polynomial time—for instance: testing whether a matroid is uniform. It might explain why it is so hard to work, at a very general level, on independence structures. It also illustrates the challenges that design science will have to face in the future.

It is important to remind that the results are obtained under the assumption that the set of techniques follows the “independence augmentation” axiom. Accepting this assumption enabled us to formulate hypotheses on the design of one technique and its impact on a set of technique and to formulate hypotheses on the dynamics of techniques. Moreover, we have shown on some empirical cases that these hypotheses are not rejected. Still this calls for further research that would help to confirm or reject the “independence augmentation” assumption and that would also help to discover new properties of C-K/Ma and test them.

Finally, relying on advanced models of design theory and matroid, this paper strengthens the study of the endogenous dynamics of technical systems and opens new ways. Beyond the phenomena and the strategy, the model might also be useful to study economics and organizational issues. In particular, the logic of “indirect” value in the design of generic technology actually raises interesting institutional issues: are companies—more interested for direct value—able to design generic technologies with high indirect value? Who could be the new actors in charge of these designs? To study the design of generic techniques, we integrated in the model the logics of cohesion and interdependences between techniques—this might now lead us to shed new light on the logics of cohesion and interdependences in economics and society.

Appendix: Definition and principles of the main notions of matroid theory used in the paper

Matroid theory	Matroid of techniques
Matroid: an ordered pair (E, I) , where E is a finite set, and I is a collection of subsets of E (which can be called the independent set of (E, I)), where I satisfies the following properties:	Elementary techniques are the elements of the matroid Working systems are the dependent set of the matroid. They are supposed to follow matroid conditions

Matroid theory	Matroid of techniques
(1) I is non-empty;	A working system is elementary if it can't be obtained by merging two smaller systems
(2) every subset of every member of I is also in I (hereditary condition);	
(3) if X and Y are in I and $ X = Y + 1$, then there is an element x in $X - Y$ such that $Y \cup \{x\}$ is in I (independence augmentation condition)	
Circuit: a subset of E that is a minimal dependent set (removing any one element of the circuit transforms it into an independent set)	A minimal working system (if a technique is removed, the working system doesn't work)
Base (or basis): a subset of E that is a maximal independent set (adding any one element to the base transforms it into a dependent set)	A set of independent techniques
Rank: corresponds to the highest size of an independent set (the size of any base)	The highest number of independent techniques
Dual of matroid: the matroid M^* formed on the same set of elements E but the bases of the duals are the complementary elements of the bases of M	Correspond to the relations between elementary working systems
Deletion (of one single element): given a matroid $M = (E, I)$ and an element e of M , deletion is the operation that forms the matroid $M - e$ by removing e from E and keeping the independent sets of I that are included in $E - e$	Removing one technique
Contraction (of one element): a deletion in the dual M^* of M . In a graphic matroid: merge two vertices linked by an edge; in a vector matroid: contraction corresponds to projection along the single element	
Extension: the operation that forms the matroid N by adding e to M such that $N - e = M$, i.e. M is the matroid resulting from the deletion of e	As shown in part 2: designing a non-pervasive technique
Coextension: an extension in the dual of M	Designing a pervasive technique (generic technique (see part 2)
A flat: a flat F is a set of elements of E such that it is impossible to add a new element of E into F without changing its rank	A set of elementary techniques that are incompatible with any other elementary technique outside the set
Connected component (2-connectedness) in a matroid M : is a matroid N of M such that for every pair of distinct elements of N , there is a circuit containing both	

References

- Arthur WB (2009) *The nature of technology. What it is and how it evolves*. Free Press, New York
- Baldwin CY, Clark KB (2000) *Design rules, volume 1: the power of modularity*. The MIT Press, Cambridge
- Braha D, Reich Y (2003) Topological structures for modelling engineering design processes. *Res Eng Des* 14(4):185–199
- Bresnahan TF, Trajtenberg M (1995) General purpose technologies: engines of growth? *J Econom* 65(1):83–108
- Dickinson HW (1936) *Matthew Boulton. Réédition de 1999 edn*. TEE Publishing, Warwickshire
- Dosi G, Freeman C, Nelson R, Silverberg G, Soete L (eds) (1988) *Technical change and economic theory*. Pinter, London
- Gawer A (ed) (2009) *Platforms, markets and innovation*. Edward Elgar, Cheltenham
- Gilles B (1986) *The history of techniques, 2 vols. Vol 1: Techniques and civilizations; vol. 2: techniques and sciences*. Gordon and Breach, New York
- Hatchuel A, Le Masson P, Reich Y, Weil B (2011) A systematic approach of design theories using generativeness and robustness. *International conference on engineering design, ICED'11*. Technical University of Denmark, Copenhagen, p 12
- Hatchuel A, Weil B, Le Masson P (2013) Towards an ontology of design: lessons from C-K design theory and forcing. *Res Eng Des* 24(2):147–163
- Hooge S, Kokshagina O, Le Masson P, Levillain K, Weil B, Fabreguette V, Popiolek N (2014) Designing generic technologies in energy research: learning from CEA technologies for double unknown management. Paper presented at the European Academy of Management, Valencia
- 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
- Kokshagina O, Le Masson P, Weil B, Cogež P (2012) Platform emergence in double unknown: common challenge strategy. Paper presented at the R&D management conference, Grenoble, France
- Kokshagina O, Le Masson P, Weil B (2013a) How design theories enable the design of generic technologies: notion of generic concepts and Genericity building operators. Paper presented at the International conference on engineering design, ICED'13, Séoul, Korea
- Kokshagina O, Le Masson P, Weil B, Cogež P (2013b) Platform emergence in double unknown (technology, markets): common unknown strategy. In: Çetindamar D, Daim T, Başoğlu N, Beyhan B (eds) *Strategic planning decisions in the high tech industry*. Springer, London, pp 90–120
- Le Masson P, Weil B (2013) Design theories as languages for the unknown: insights from the German roots of systematic design (1840–1960). *Res Eng Des* 24(2):105–126
- Le Masson P, Hatchuel A, Kokshagina O, Weil B (2015a) Generic technique and the dynamics of technologies: using matroid and design theory to design techniques with systemic impact. In: *International conference on engineering design, Milan*
- Le Masson P, Weil B, Kokshagina O (2015b) A new perspective for risk management: a study of the design of generic technology with a matroid model in C-K theory. In: Taura T (ed) *Principia designae—pre-design, design, and post-design—social motive for the highly advanced technological society*. Springer, Tokyo, pp 199–219
- Le Masson P, Hatchuel A, Weil B (2016) Design theory at Bauhaus: teaching “splitting” knowledge. *Res Eng Des* 27:91–115
- Neel DL, Neudauer NA (2009) Matroids you have known. *Math Mag* 82(1):26–41
- Oxley J (2011) *Matroid theory*. Oxford graduate texts in mathematics, 2nd edn. Oxford University Press, Oxford
- Reich Y (1995) A critical review of general design theory. *Res Eng Des* 7:1–18
- Saviotti PP, Metcalfe JS (1991) *Evolutionary theories of economic and technological change*. Harwood Academic Publishers, Newark
- Shai O, Reich Y (2004a) Infused design: I theory. *Res Eng Des* 15(2):93–107
- Shai O, Reich Y (2004b) Infused design: II practice. *Res Eng Des* 15(2):108–121
- Simondon G (1958) *Du mode d'existence des objets techniques. L'invention philosophique, 3ème édition de 1989 edn*. Aubier
- Strumsky D, Lobo J (2015) Identifying the sources of technological invention. *Res Policy* 44(8):1445–1461
- Suh NP (1990) *Principles of design*. Oxford University Press, New York
- Suh NP (2001) *Axiomatic design: advances and applications*. Oxford University Press, Oxford
- Thurston RH (1878) *A history of the growth of the steam engine*. Appleton, New York
- Ulrich KT, Eppinger SD (2008) *Product design and development*, 4th edn. Mc Graw Hill, New York
- Whitney H (1935) On the abstract properties of linear dependence. *Am J Math* 57(3):509–533
- Yoshikawa H (1981) General design theory and a CAD system. In: Sata T, Warman E (eds) *Man-machine communication in CAD/CAM, proceedings of the IFIP WG5.2-5.3 working conference 1980 (Tokyo)*. North-Holland, Amsterdam, pp 35–57