INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED13
19-22 AUGUST 2013, SUNGKYUNKWAN UNIVERSITY, SEOUL, KOREA

HOW DESIGN THEORIES ENABLE THE DESIGN OF GENERIC TECHNOLOGIES: NOTION OF GENERIC CONCEPT AND GENERICITY IMPROVEMENT

Olga KOKSHAGINA (1,2), Pascal LE MASSON (1), Benoît WEIL (1), Patrick COGEZ (2)
1: Mines ParisTech, France; 2: STMicroelectronics, France

ABSTRACT

Generic technologies enable to create benefits across wide range of industrial applications. Though providing important insights on generic technologies commercialization, less attention was paid to generic technologies creation. Then, is it possible to design directly generic technologies? Can the intention to build genericity be expressed ex ante? The proposed study indicates that formal design theories provide powerful mechanism of genericity construction when the environments are initially fixed and partially unknown. It is demonstrated that starting point to design generic technology is a generic concept. In addition to a concept definition proposed by C-K theory, the descriptors associated to the domain of existence were added. The generic concept targets the existence domains that are not reduced to one solution but several of them both known or partially unknown. Moreover, it is highlighted that different descriptors of existence domain can lead to various genericity levels. The economic reasoning behind the genericity building provides insights on the dynamics of engineering systems.

Keywords: design theories, generic technology, genericity, generic concept, genericity building, existence domain, descriptors of existence domain

Contact:
Olga Kokshagina
Mines ParisTech
Center for Management Science CGS
Paris
75272
France
olga.kokshagina@mines-paristech.fr

1 INTRODUCTION

Technology management stresses the importance of generic technology (GT) development. Numerous studies have attempted to understand the value of general-purpose technologies from historical (Landes, 1990, Moser and Nicholas, 2004), economical (Lipsey et al., 2005), sociological perspectives (Powell, 1987). The investment in GT development is recognized as one of the instruments to deal with high marketing and technological uncertainty. Successful examples of GTs exploration are demonstrated through the analysis of electricity, semiconductors, steam engines, nanotechnologies (Edquist and Henrekson, 2006, Rosenberg and Trajtenberg, 2004). GT signifies breadth of application areas (Bresnahan and Trajtenberg, 1995) and is characterized by its technological dynamism. To account for GT, one has to design a platform that by definition incorporates the underlying core technology and facilitates development of a stream of market derivatives (Gawer, 2009). These platforms support generic technologies development, which create leveraged growth, market variety and high benefits. Various scholars investigated the impact of GT development *ex post* (Edquist and Henrekson, 2006). Providing important insights on GT commercialization (Maine and Garnsey, 2006), market complementarities development, less attention was paid to generic technology creation.

The goal of this paper is generic technology creation and more precisely genericity building. Following evolutionary approaches of technological change (Nelson and Winter 1982, Dosi 1982) that deal with path dependencies and consequent trial exploration, GT emergence can be achieved using trial and error mechanisms (Miyazaki, 1994). Generic platform development can be build upon a trial, where trials are often selected from the known list of alternatives. Then, is it possible to directly design generic platforms? Can the intention to build genericity be expressed *ex ante*? The aim of the current work is to understand whether there exists a design process aiming to build genericity.

Dealing with the issue of GT design, formal design theories are used in the paper. The study shows that formal design theories provide powerful mechanism of genericity building when the environments are initially fixed or still unknown. Literature review enables to highlight the importance of descriptors of existence domain to construct genericity in unknown. The introduced generic concept targets the existence domains that are not reduced to one solution but several of them that can be known or partially unknown. To demonstrate the possible extensions of applications range, different forms of domain of existence are used: the formal mathematical models and engineering models. The most recent C-K Design Theory is used to illustrate genericity building.

2 LITERATURE REVIEW: DESIGN FOR GENERIC TECHNOLOGIES: GENERICITY AND GENERICITY BUILDING

2.1 Generativity and genericity of design theories

Then, what is genericity and how formal design theories account for genericity building? Formal design theories aim to reach mathematical and logical rigor both to formulate hypothesis and to prove their findings (Hatchuel et al., 2011). The following theories were selected for the study: General Design Theory, Axiomatic Design, Coupled Design Process, Infused Design, Concept-Knowledge design theory. It was shown that design theories are theories of generativity (Hatchuel et al., 2011). Generativity is defined as a capacity to describe the set of various alternatives; it is an "ability to produce novel design proposals". Design theories had to think beyond pure combinatorial processes and explore different forms of generativeness: dynamic transformations, adaptations, hybridizations, discovery and renewal of objects (Hatchuel et al., 2011). Generativity enables new objects creation with desired properties and ensures new knowledge exploration for their existence.

Genericity seeks to integrate economic reasoning, enables the economy of knowledge expansions for new objects creation. It aims to obtain several solutions at once. Genericity is defined as a capacity to propose generic technological core that address set of environments in generative space. Genericity can be defined over generativity. Then, how was genericity tackled by design theories so far?

2.2 Design Theories and genericity

The design theories don't explicitly discuss technologies and genericity, but design solutions (Pahl and Beitz, 1995). In certain design theories, one design solution has to be robust to several known environments. The several environments, for which a solution has to be validated, is called here an existence domain. There are certain forms of genericity and technologies. In order to distinguish these notions of genericity and technology treated by each theory, the analysis of formal models is needed.

In the case of **General design theory (GDT)** (Yoshikawa, 1985) a design solution is an entity included in specifications and contains necessary manufacturing information (Reich, 1995). Entities are characterized through subsets that are either functions or attributes. The design process consists in designating a domain on the attribute concept space, which is included by that on a concept function space. The design process is limited to the combination of prescribed functions and structures following axioms of GDT. According to the theory, if the entity space has Hausdorff structure, we can address all the functional combinations. The problem appears when the space of attributes doesn't cover completely the set of functional combinations. Then Hausdorff correspondence is impossible to construct and all the functional combinations cannot be addressed.

It can be interpreted as a form of genericity associated to the set of functional combinations. The technology can be considered as the structure of attribute space available to address the function space in Table 1. The structure of entity space defines genericity. When structure of entity space has Hausdorff structure (ideal knowledge), maximum genericity can be obtained. If not, genericity can be conceived only according to certain subset of functions.

In **Axiomatic Design (AD)** (Suh, 1999) the design process occurred in physical domain and functional domain. The goal of the designer is to built mapping between functional requirements (FRs) in the functional domain and design parameters (DPs) of the physical domain by selecting proper DPs to ensure FRs. In the axiomatic approach design is controlled by two axioms of Suh: the independence and information axiom. By obtaining the decoupled diagonal matrix of FRs and DPs, one can address each function separately or developed DPs can be changed according to each function independently. If a matrix is decoupled then the designers will be able to reach all the functional combinations. So in AD genericity is constructed on the set of achievable functional requirements. The technology can be defined as the set of FRs and DPs relations in Table 1. The higher genericity is obtained when the relations in between DP and FRs are ensured by independency axiom and the matrix is diagonal.

Both GDT and AD provide tools that are quite powerful to design GT. The form of genericity does not pose any problems since the environments are considered to be known and predefined in advance by the entity space or functional combinations. Genericity is described by the extension of the existence domain, which is defined by the set of all possibly described environments.

The **Coupled design theory (CDP)** (Braha and Reich, 2003) offers the way to organize design process of functions and descriptions co-evolution by transition in 'closure' spaces. Creating functional or structural closures at each design step enables to integrate new knowledge and consider functions emergence from 'closure' spaces. It means that in between functions there are order relations. This theory formalizes exploration in partially unknown spaces and closure operators lead to conceive not just existing functions but also their proximal refinement. CDP enables to treat genericity beyond initial functional combinations. Genericity can be explained as functions and new words - future functions candidates. The higher genericity is achieved thanks to closure operators. CDP incorporates unknown environments by consideration of new words and provides the attributes to facilitate the generic concept definition. This theory enables to deal with **genericity induced by unknown** in Table 1. Yet, designed process is limited by pre-established closure spaces. The operators of transitions in closure spaces ensure genericity, but the exploration of new functional sets is limited by proximal closure operations. Technology is described as functional and structural descriptions.

Infused design (ID) (Shai and Reich, 2004) permits not only to create functions and organize explorations in closed space proximity like CDP, but to integrate new knowledge from other disciplines. In ID instead of looking for closure functions, the goal is the common representation composed of model type and terminology (M, T) that accommodates all the original ones. The ID privileges transformations of original into reference problems. These transformations are based on mathematical operations of duality, generality, and equivalence.

ID shows the possibility to construct genericity and not just to account for it. In ID the genericity can be defined as combination of classical genericity inside each problem and the form of genericity in between them. The first type of genericity is similar to the ones in CDP, AD and GDT. The second is ensured by transformation operators and brings the power of knowledge generation. For instance, duality operators enable the propagation of the previously unknown solution in between different domains. The nature of results is not known a priori. There is a form of **genericity induced by unknown that relies several knowledge domains**. Genericity construction is based on the operators and not defined over space. The existence domain is defined by the operators of propagation.

The knowledge circulation can provoke both new solutions and also new surprises and concepts (like face force discovery). The technology is the set of model type and terminology in each and across domains. Then, the genericity gain is ensured by obtaining generic solutions in each domain and their transfer across the disciplines using operators.

Table 1.Technology and genericity definitions in Design theories

	GDT	**AD**	**CDP**	**ID**	**C-K**
Technology	Structure of entity space	Set of FRs and DPs relations	Functional and structural descriptions	Set of models and terminology in each and across domains	Initial knowledge (K-space)
Genericity defined over extension domain	Functional combinations	Range of functional requirements	Functional combinations and new words	Within problems (discrete models) and duality across problems	K-reordering, Concepts
Genericity Improvement	Hausdorff measure	Diagonal matrix (independency)	Closure operations	Transformation operations	***Descriptors of existence domain***
Genericity by reference to a fixed referential			**Genericity induced by unknown**		
Defined over Space				**Genericity operators**	

Concept – Knowledge (C-K) design theory (Hatchuel and Weil, 2009) doesn't use mapping in between functions and attributes. It defines the design process as a continuous refinement of a concept described by various properties P_i that need to be met based on existing knowledge and producing new one. The design reasoning is based on the distinction and interaction in between two independent spaces: Concept (desired unknown) and Knowledge space (existing and available knowledge). In C-K the concept is defined as « There exists some object x for which a group of properties P_1, P_2,..., P_n are true in K » (Hatchuel and Weil, 2009) such that a concept is undecidable with respect to current *K*. According to C-K theory, creative design requires an expansive paritition. This partition will enable creation of new attribute(s) *P'* that was not initially an attribute of *x* in *K*. This theory models the integration of new knowledge and connections in between knowledge disciplines through operators of knowledge disjunction (K -> C), conjunction (C ->K), and expansion (C ->C, K->K) in Figure 1.

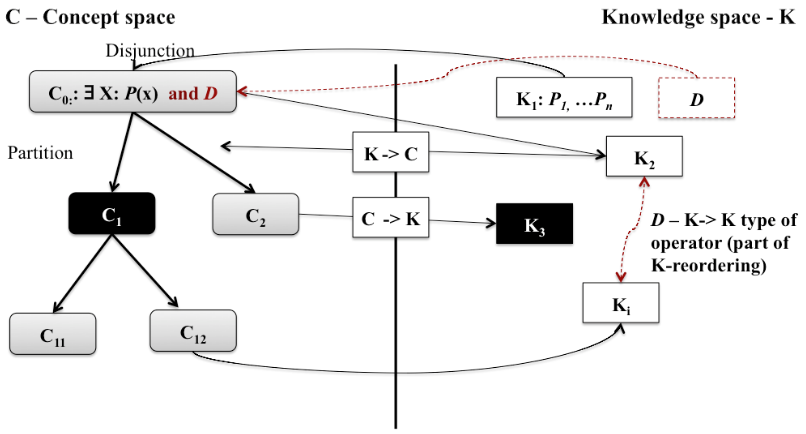

Figure 1: Generic extension in C-K theory

In C-K technology can be defined as the initial K-space. Genericity goes beyond the validation of *ex ante* targeted sets of specifications. It includes all the new objects generated by K-reordering (as well as new concepts), i.e. by combination of the newly designed object with the previously known objects, the combination of the new and the old knowledge. As in ID, genericity is based on genericity building operators. More precisely in C-K theory it depends on K-reordering, which is defined as the operation of propagation that follows the conjunction emergence (C ->K): this is the effect of the newly generated knowledge, coming from the unknown, on the K space. Genericity improvement in C-K appears to be linked to the expansion strategy: certain design can enable limited K-reordering – object identity preservation or the complete object reconstitution – object identity discussion.

To summarize, the existing theories give a tendency to predefine generic space that can be constructed through combination of known and unknown. The different types of genericity were identified: genericity by reference to a fixed referential in known and genericity induced by unknown. In case of known environments, GDT and AD provide powerful solutions. For genericity in unknown, it has been demonstrated that the design theories evolution brings the possibility to reason on unknown yet environments (CDP, ID and C-K). Genericity is defined over existence domain and there are various forms of descriptors of existence domain for genericity building. Genericity can be analyzed as a Hausdorff measure in an entity space (GDT), as an hypercube in vector space of FRs (AD), as a set of closure operators (CDP) in Table 1. The most generative formal models lead to distinguish genericity defined over a fixed referential (entity space, functional space, closure space) or obtained by operators that help to propagate and hybridize the new into the old, beyond the new validated conjunction. These descriptors can be 1) deduced from existing knowledge (as in case of duality operator in ID) meaning that the rules of propagation exist and are used for knowledge propagation or 2) can be conceived during the exploration. Literature review reveals different genericity building descriptors but their contribution to acquired genericity can be demonstrated only when the results are obtained. Then, how these descriptors of existence domain influence genericity construction *ex ante*?

To tackle this question, it is important to properly introduce the notion of generic concept itself. A concept is called generic once it is valid for several environments. Obviously it should have an existence domain beyond a singleton and lead to explore multiple application areas. GT creation is often considered as progressive trial and error mechanism of genericity building. But can a generic concept be identified *ex ante* and how to ensure that it leads to GT creation?

In C-K if an initial concept $xP(x)$ is valid for multiple applications, the descriptors of existence domain (called here D) are already established. Following AD and GDT this concept is generic for multiple known applications. In ID and C-K there could be descriptors of multiple environments D, which, combined with the existing knowledge, could lead to new applications creation. Then, what is the form of D that leads to genericity creation? Can one account for genericity creation by adding D to the formulation of the initial concept: $xP(x)D$, Figure 1. It leads to formulate following research questions:

Q1: What is generic concept *ex ante*? How does genericity depend on the descriptors of existence domain D? Furthermore, there should be a possibility to increase genericity.

Q2: How to account for higher genericity building?

To tackle identified research questions on genericity induced by the unknown, the Concept-Knowledge Design theory (Hatchuel and Weil, 2009) is used since it is independent from a particular engineering domain. The question of genericity improvement is tackled and different possibilities are indicated.

3 GENERIC CONCEPT AND GENERICITY BUILDING

The aim is to understand the mechanisms of expansive genericity. The work is conducted on modeling the necessary descriptors of existence domain. Following the previously defined hypothesis, a generic concept can be presented as $xP(x)D$. In C-K theory when one particular conjunction in space C is achieved, a new entity can be created P'. In K space propagation operators need to establish the relations in between P' and all the other knowledge entities $K_1, K_2,...,K_n$ using the descriptors of existence domain $d_1, d_2,...,d_n$ included in D. From one conjunction plenty of them can be obtained.

Example: For instance, while designing a new car breaking system for the new Volkswagen XYZ, the existence domain is reduced to one type of car and obtained genericity is zero. The rules of compatibility of a breaking system have to be compliant with Volkswagen XYZ and thus, considered to be known (D exists in K). Conversely, if one seeks to design a new breaking system that can be substituted to all the existing or not yet created car architectures, the compatibility rules are unknown. They have to be conceived during the exploration process (D is unknown).

Thus, generic power of concept is based on the association it creates in between new object and the elements of K basis. The two cases are examined: 1) $D \in K$ exists in initial knowledge basis, the descriptors of existence domain are fixed 2) $D \notin K$ has to be designed and is unknown in K.

3.1 Genericity building in case of existing descriptors of existence domain

To study the logic of genericity construction in case of existing descriptors of existence domain, the cases with mathematical and engineering knowledge structures are investigated.

Illustration 1: Mathematical knowledge structure – algebraic extensions

In mathematics various extension operators could be interpreted as design (Hatchuel et al., 2012). The main idea of field extensions is to start with a base field and construct a larger field that contains the base field and satisfies additional properties. In case of algebraic extensions knowledge basis K is defined and propagation rules are ensured by mathematical operations of addition and multiplication in Figure 2. To extend the field Q of rational numbers, the initial concept type $\exists$X: $f(x) = 0$ has to be solved. Variety of partitions can be obtained ensuring generativity of design space. For instance, taking $x^2-2=0$, a polynomial of Q with rational coefficients. The roots of $x^2-2 =0 \not\in$ Q – hence, there is an "unknown" object to conceive. It is possible to solve this equation by functional approximation. This algorithmic method allows to consequently build a $\sqrt{2}$ as a limit of a series of rationales. At the end a new $xP(x)$ solution is obtained – $\sqrt{2}$. Its creation doesn't enable its compatibility with the field Q**,** since in Q there is a possibility to multiply and addition rational numbers only.

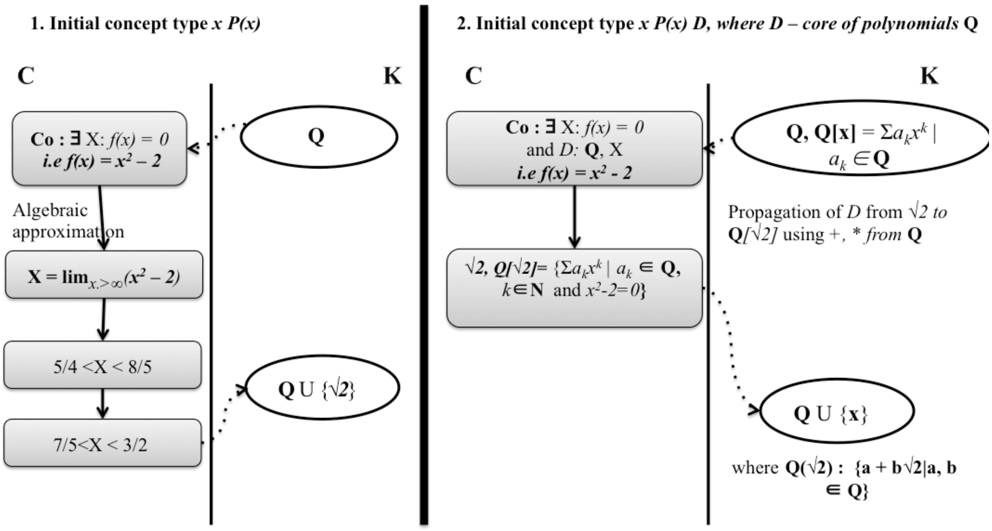

Figure 2. Algebraic fields expansion in C-K-theory

Following the theory of algebraic extensions (Bastida, 1984), another solution would be to find a non-zero polynomial type $\Sigma a_k x^k$ with $a_k \in$ Q. The concept $\exists$ X:$f(x) = 0$ contains directly D – a core of polynomials **Q**[x]= $\Sigma a_k x^k$ with $a_k \in$ **Q.** The algebraic extensions lead to work on a concept $xP(x)D$.

For instance, direct consideration of $\sqrt{2}$ from the field of real numbers R, enables to obtain solution. The obtained $\sqrt{2}$ can be propagated to new entities by combining the old Q with the new $\sqrt{2}$ through operations preserved from **Q**[x]. The descriptors D in this case are operators of addition and multiplication. It is important to underline that operators of multiplication and addition used to ensure knowledge propagation from Q to R exist in initial knowledge basis and preserved. Generic concept is obtained through propagation from $\sqrt{2}$ to **Q**[$\sqrt{2}$] using these combinative operations. In the case of algebraic extension, descriptors of extension domain D are given at the beginning. For instance, this propagation also generates similar entities. For instance x^2-1 is equivalent to 1 because $x^2-1= x^2-2 +1$ and $x^2-2=0$ or equivalent to x^4-4x^2+5 as $x^4-4x^2+5 = (x^2-2)^2+1$.

Adding directly an extension field **Q**[x] to the field of rational number forms both a new solution $\sqrt{2}$ but also all the finite extensions of the field Q type Q{$a+b\sqrt{2}$| $a,b \in$ Q}. The conjunction is made once all the rules of propagation are obtained. When D is added to the formulation of initial concept, it enables to obtain the set of solutions and not just a new number $\sqrt{2}$.

Illustration 2: Engineering knowledge structure – Steam engines development

To investigate the generic concept identification with engineering knowledge structure, steam engines example is chosen. Steam engine is recognized as a general purpose technology (Rosenberg and Trajtenberg, 2004) used in various domains as reciprocating pumps, spinning mules, driving machinery in factories, mines, propelling transports.

In 1712 Thomas Newcomen dealing with the increasing need in energy, developed atmospheric engines that brought benefits to mills, but they were inefficient in terms of the energy use to power them. Handling this problem of energy loss, James Watt developed a steam engine with separate condensation chamber (1763). The main use was to pump water out of mills. In 1780s Watt's business partner, Matthew Boulton, an entrepreneur aware of the issues of the newly emerging industry (in particular iron work, through his collaboration with Wilkinson), had an idea to develop an energy source compatible with industrial systems like textile factories, driving machinery in Figure 3.

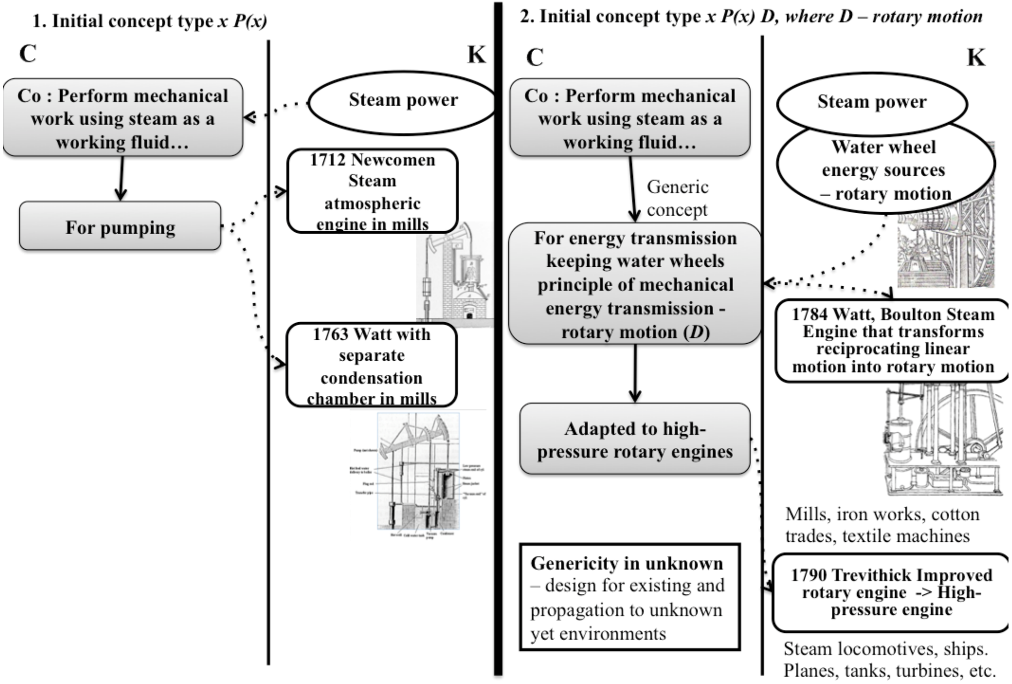

Figure 3. Generic steam engine design in C-K-theory

According to (Dickinson, 2010) "Boulton created the environment which enabled Watt to work on the further instrument that brought steam engines from an apparatus for lifting water into one with an immensely larger field of application to general power purposes". He conceived existing knowledge on commonly deployed water wheel energy sources to ensure rotary motion between energetic source and the mechanism that uses this motion. In 1781 Boulton wrote to Watt: "we should determine to take out a patent for certain methods of producing rotative motion" (Dickinson, 2010). The conversion of reciprocating into rotary motion was facilitated by Watt's development of double acting engine which resulted in a more uniform movement of the piston and made this design state of the art for applications with rotary motions (Frenken and Nuvolari, 2004).

The generic concept incorporated *D* based on the preserved system of water wheels energy transmission in Figure 3. The design of a set of entities was achieved through the identification of one descriptor of existence domain – rotary motion (*D*). This concept is generic since it is compatible both with known environments, where the energetic source can be transferred by rotary motion, and with the extensive number of unknown initially environments. Furthermore, Boulton and Watt established standard units of measure for both the fuel efficiency (duty) and the power (horsepower), which lead to further diffusion and economic significance of engines. It is important to underline the effort on genericity adaptation. Once steam engine with rotary motion was designed, it took years to improve and disseminate them to the markets. E.g., first locomotive with steam power was introduced in 1804, 20 years after the invention of the first rotary engine. These examples demonstrate that what counts is the capacity to enable propagation, fabricate sets of the relations in between objects. Both algebraic extension and rotary steam engines use existing descriptors of existence domain to design new objects.

3.2 Genericity construction in case of designed descriptors of existence domain

In the second case $D \not\subseteq K$ has to be conceived and is unknown in *K*. Then, how to obtain a generic concept in case of unknown descriptors of extension domain?

Illustration 1. Mathematical knowledge structure - complex numbers

The process of genericity building described in **3.1** stops when there are no more solutions for $\exists X$: $f(x) = 0$ beyond the initial knowledge set. Consider $f(x) = x^2+1 \in$ Q(x). It is impossible to use fields of polynomials **Q**[x] as *D*. x^2+1 =0 has no root in R, it is a maximal ideal (Bastida, 1984). To extend it, one has to use the field of **R**[x]. Working on a closed field R, Cauchy (1831) extended Sturm's theorem to count and locate the real roots of any real polynomial to count and locate the complex roots of any real polynomial. Created by Cauchy extension can be interpreted as a descriptor *D* that leads to complex extension C = R$[i]$, where $i^2 = 1$. The roots of $f(x)$ are $+/-i$, and so $f(x)$ splits in C; that is, $f(x) = (x+i)(x- i)$ is a product of linear polynomials in **C**[x].

In this case the logic of *D* renewal enables to build higher genericity that incorporates the field of complex numbers. It is impossible to obtain solution to $f(x) = 0$ which has no root in C. Following logic of polynomial operators, the maximal genericity is obtained.

Illustration 2. Engineering knowledge structure – MEMS case

MEMS (Micro Electro Mechanical System), micron-size devices that can sense or manipulate the physical world, are exceptionally diversified. MEMS encompass the process-based technologies used to fabricate tiny integrated devices and systems that incorporate functionalities from different physical domains into one device. MEMS revolutionized various product domains and created new ones by bringing together silicon-based microelectronics and micromachining technology (Bryzek, 1996).

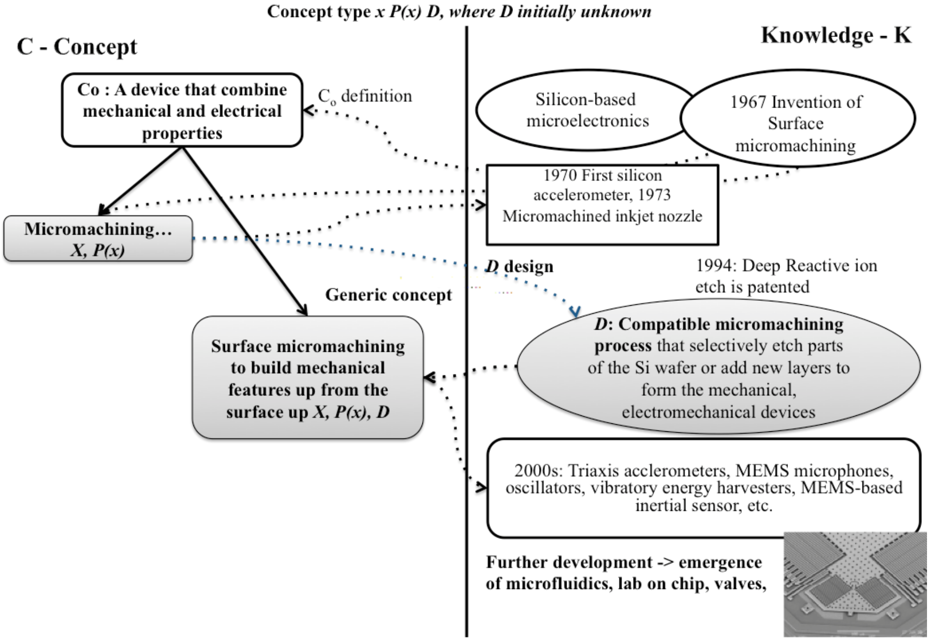

Figure 4. Generic MEMS technology and D design in C-K-theory

In the last decade, MEMS have provoked revolutions in several industries: arrays of micromirrors enabled digital film projectors, accelerometers like those in Wii controller have changed gaming, the iPhone 4 became the first portable consumer device to incorporate a three-axis accelerometer, three-axis gyroscope, and three-axis electronic compass. MEMS became a generic technological platform that enables the development of products, augmenting the computational ability of microelectronics with the perception and control capabilities of microsensors and microactuators, expanding the space of possible designs and applications. MEMS technology is the integration of mechanical elements, sensors, actuators, and electronics on a common substrate through microfabrication technology (*D*).

MEMS development started in 1950s when silicon strain gauges became commercially available. Not far after this, Dr. Richard Feyman, in his famous talk "There's plenty of room at the bottom", described the tremendous potentials in the micro technology field. After the invention in 1967 of surface machining the first silicon accelerometer, micromachined inkjet nozzle were developed (Heeren and Salomon, 2007). In the mid 1990s two enabling technologies appeared. The deep reactive ion etching of silicon made possible to etch deep high aspect ration trenches into silicon. The development of silicon on insulator wafers enabled high quality silicon layers for micromechanical structure. Thanks to these technologies, the micromechanical components were fabricated using compatible micromachining processes to selectively etch away parts of the silicon wafer or add new structural layers to form the mechanical and electromechanical devices (*D* design). Through these innovative processes MEMS became a true GT and manufacturing platform. Thus, MEMS appeared to be generic when new descriptor *D* that ensures its propagation emerged. The genericity continued to be built. For instance, the emergence of microfluidics in the medical applications opens a lot of possibilities for MEMS in the drug delivery in Figure 4.

4 GENERICITY IMPROVEMENT

While pursuing genericity, the goal is to maximize the number of environments GT will address. Then, how to account for higher genericity?

Illustration 1. Mathematical knowledge structure – Field extensions

Going back to the example of algebraic field extension, once new field $Q(x_1)$ is designed, it can be spread in $Q(x_1)$ (x_2), … these fields are consequently added - the new one is "bigger" than the previous with the condition that $f(x_n) = 0$ has no root in $Q(x_n\text{-}1)$. So having $Q(\sqrt{2})$, it is possible to proceed with X2-$\sqrt{2}$=0 which doesn’t have roots in $Q(\sqrt{2})$ and generates $Q(\sqrt{2})(2^{\wedge}1/4)$ which is higher in genericity. The “size” of the extended field depends on the initial concept. For instance, consider the polynomial $x^4 - 2 = 0$ over Q. It roots are ${}^4\sqrt{2}$, $-{}^4\sqrt{2}$, ${}^4\sqrt{2}i$, $-{}^4\sqrt{2}i$. x^4-2 has splitting field $Q(\sqrt[4]{2}, i)$ of the size 8 over Q. To write every element of this field, one need 1, ${}^4\sqrt{2}$, ${}^4\sqrt{8}$, ${}^4\sqrt{2}i$, ${}^4\sqrt{4}i$, ${}^4\sqrt{8}i$, i and so the field size is 8. Based on the root $\sqrt[4]{2}$, the generated field is $Q({}^4\sqrt{2})=\{a+b\,{}^4\sqrt{2}+c^{4/3}\sqrt{2} + d\sqrt{2}$, a, b, c $\in$ Q}, where $Q({}^4\sqrt{2})/Q$ has size 4. There is $Q \subset Q(\sqrt{2}) \subset Q({}^4\sqrt{2})$. The extension will also depend on the K base. In Q, it is possible to provoke infinity of more or less generic extension. But in R, it is proven that all algebraic extensions of R are isomorphic to the field of complex numbers C.

Illustration 2. Engineering knowledge structure – steam engines

Drawing on the principles of engineering design, the design process can be accomplished through: planning and clarifying the task, conceptual design, embodiment design, detail design (Pahl and Beitz, 1995). Theoretically, it is possible to account for genericity improvement at each phase till the detail stage of design is achieved. Embodiment keeps technical specification and the goal is to ensure predefined volumes, which means to ensure compatibility with the identified environments. These are descriptors (*D*) defined at the level of embodiment (based on design parameters).

In case of steam engines the compatibility with factory machinery was achieved through rotary motion preservation. Using the principles of engineering design language, the descriptors are defined at the embodiment level. Though, the existence domain can be defined at the conceptual or functional level as well. For instance, one could think of designing steam engines compatible with various forms of energy transmission. In this case, descriptors should be defined at the functional level.

Example: New automotive engine can be generic by being substitutable to the previous engine (keep the transmission logic, the peripheral components). In this case all interfaces, are kept equivalent (*D* preserved) and genericity improvement is based on embodiment. Otherwise, new automotive engine will require to create new interface rules to be compatible with a range of vehicle (the electric motor might lead to redesign the interface of air-conditioning, safety systems), incorporate other functions.

Thus, genericity improvement is at the core of engineering language and can be achieved by functions or by embodiment.

5 RESULTS AND DISCUSSION

The evaluation of GT is often based on the number of markets it achieves, the benefits it generates. From the engineering point of view the genericity is characterized by the number of potential applications that technology will address, which requires reasoning in unknown.

According to the first research question, the proposed study demonstrated that to be generic *ex ante* a concept has to incorporate descriptors of existence domain *D*. A generic concept has a form *XP(x)D*, where *D* ensures the composition of *x* with the existing knowledge domains and bring forward knowledge propagation towards unknown environments. It is demonstrated that generic concept aims at designing not only one entity but aims at generating a set of entities. The descriptors of existence domain *D* can be identified from the initial knowledge or conceived during the design process. Based on the knowledge structure they can take various forms: it can be mathematical operators in case of field extension or they can be associated with the compatibility of the object and associated environments in case of engineering knowledge. In case of genericity construction in unknown, the goal is not to obtain a new object through expansive partition but a new object that influence the existing knowledge domains and create maximal knowledge propagation. Having defined a form of *D* imposes a certain structure of future knowledge and control the operation of K-reordering, which brings interesting theoretical perspective to C-K theory. Regarding the second question, the level of attained genericity can be improved. In the mathematical knowledge structure the measure of the field size characterizes genericity level. In the real objects the measure is the number of partially unknown environments compatible with the emerging object through descriptors of existence domain.

This work is limited to the study of two forms of genericity based on algebraic field extensions and selected engineering objects. Other knowledge topologies could enable to identify new powerful form of genericity. For instance in models of K-space (Hendriks and Kazakci, 2010) emphasized that different structures to model knowledge yield different conceptive power and degrees of flexibility. Moreover, not examined in this work social and psychological design approaches could provide new perspectives on the process of GT design. The economic reasoning behind the genericity building provides insights on the dynamics of engineering systems.

REFERENCES

Bastida, J. R. 1984. *Field extensions and Galois theory*, Cambridge University Press.

Braha, D. & Reich, Y. 2003. Topological structures for modeling engineering design processes. *Research in Engineering Design,* 14**,** 185-199.

Bresnahan, T. F. & Trajtenberg, M. 1995. General purpose technologies 'Engines of growth'? *Journal of econometrics,* 65**,** 83-108.

Bryzek, J. 1996. Impact of MEMS technology on society. *Sensors and Actuators A: Physical,* 56**,** 1-9.

Dickinson, H. W. 2010. *Matthew Boulton*, Cambridge University Press.

Edquist, H. & Henrekson, M. 2006. Technological breakthroughs and productivity growth.

Frenken, K. & Nuvolari, A. 2004. The early development of the steam engine: an evolutionary interpretation using complexity theory. *Industrial and Corporate Change,* 13**,** 419-450.

Gawer, A. 2009. *Platforms, Markets and Innovation*, Edward Elgar Publishing.

Hatchuel, A., Le Masson, P., Reich, Y. & Weil, B. A systematic approach of design theories using generativeness and robustness. Proceedings of the 18th International Conference on Engineering Design (ICED11), Vol. 2, 2011. 87-97.

Hatchuel, A. & Weil, B. 2009. CK design theory: an advanced formulation. *Research in Engineering Design,* 19**,** 181-192.

Hatchuel, A., Weil, B. & Le Masson, P. 2012. Towards an ontology of design: lessons from C–K design theory and Forcing. *Research in Engineering Design***,** 1-17.

Heeren, V. H. & Salomon, P. 2007. MEMS Recent Developments, Future Directions. Loughborough University.

Hendriks, L. & Kazakci, A. 2010. Design as imagining the future: a formal account. Logic and Interactive Rationality. *In:* Grossi, D., Minica, S., Rodenhauser, B. & Smets, S. (eds.) *Logic and Interactive Rationality.* Yearbook2010.

Landes, D. S. 1990. Why are we so rich and they so poor? *The American Economic Review,* 80**,** 1-13.

Lipsey, R. G., Carlaw, K. I. & Bekar, C. T. 2005. *Economic Transformations: General Purpose Technologies and Long-Term Economic Growth,* OUP Oxford.

Maine, E. & Garnsey, E. 2006. Commercializing generic technology: The case of advanced materials ventures. *Research Policy,* 35**,** 375-393.

Miyazaki, K. 1994. Search, learning and accumulation of technological competences; the case of optoelectronics. *Industrial and Corporate Change,* 3**,** 631-654.

Moser, P. & Nicholas, T. 2004. Was electricity a general purpose technology? Evidence from historical patent citations. *American Economic Review***,** 388-394.

Pahl, G. & Beitz, W. 1995. *Engineering design: a systematic approach*, Springer.

Powell, W. W. 1987. Explaining technological change. JSTOR.

Reich, Y. 1995. A critical review of general design theory. *Research in Engineering Design,* 7**,** 1-18.

Rosenberg, N. & Trajtenberg, M. 2004. A general-purpose technology at work: The Corliss steam engine in the late 19th century United States. *The Journal of Economic History,* 64**,** 61-99.

Shai, O. & Reich, Y. 2004. Infused design. I. Theory. *Research in Engineering Design,* 15**,** 93-107.

Suh, N. P. 1999. A theory of complexity, periodicity and the design axioms. *Research in Engineering Design,* 11**,** 116-132.

Yoshikawa, H. 1985. Design theory for CAD/CAM integration. *CIRP Annals-Manufacturing Technology,* 34**,** 173-178.