INTERNATIONAL DESIGN CONFERENCE - DESIGN 2010
Dubrovnik - Croatia, May 17 - 20, 2010.

SIMULATION OF DESIGN REASONING BASED ON C-K THEORY: A MODEL AND AN EXAMPLE APPLICATION

A.O. Kazakci, A. Hatchuel, P. Le Masson and B. Weil

Keywords: design reasoning, simulation, C-K design theory

1. Introduction

The paper presents a simulation model called Graphi.CK that is based on the framework provided by C-K design theory [Hatchuel and Weil 2007, 2009]. C-K theory is a general theory of design *reasoning*. It suggests a fundamental distinction between *concepts* (propositions about novel objects) and their interaction with the *knowledge* (propositions about known objects) of the designer. According to the theory, the interaction and co-evolution of concepts and knowledge is the main engine through which design progresses.

While C-K theory has been developed in connection with several empirical investigations [Le Masson, Weil et al. 2006], it foundations are rooted in modern set theory [Hatchuel and Weil 2009]. Most of the formal work about C-K has focused on such logical and mathematical aspects. Hatchuel and Weil (2007) propose a key insight establishing a parallel between the forcing technique (a technique used to create new set-theoretical universes) and C-K theory. Kazakci et al. (2008) suggest that the core ideas of C-K theory can be modelled or implemented independently of set theory. They introduce a notion of *models of K space* and they show that graph-based term logic may be used to interpret all the key notions of C-K theory. [Kazakci 2009] presents a logical model of C-K theory based on the constructivist logic of Intuitionism.

By contrast to these work focusing on logical and mathematical modelling of C-K theory, this work suggest *a simulation-based approach* as an alternative way of further investigation of the theory and the class of design phenomena it describes. Beyond practical tools, simulation based inquiry has been recognized as valid scientific method in many fields ranging from economics to social sciences. It can be seen as an intermediary step in the scientific progress for generating models and testing hypothesis with *dry lab* experiments [Wolfram 2002]. Simulation models for studying the nature and the impact of design have also been used in design research. Sosa and Gero (2002, 2003) studied the impact of design on social change. Saunders and Gero (2002) proposed a model of curious agents reporting evaluations of designs. Taura and Nagai (2008) propose that recognizing differences is a key element in design reasoning and they suggest simulation studies as a means for analyzing it.

The current study presents a contribution to the simulation-based study of design. We describe a model of C-K design theory, named *Graphi.CK,* enabling simulations of design reasoning. The model uses a basic graph- based knowledge representation. A software implementation of the model has been developed using Java and JUNG (Java Universal Network Graph) library. The two main objectives of Graphi.CK are (i) to offer pedagogical means in teaching C-K theory using visualization capabilities of JUNG and (ii) investigating various hypothesis and design strategies – here we concentrate on this latter aspect.

Engineering design vs. engineering science: a discussion of a real case based on Graphi.CK

We discuss the model and some results with respect to the case of an industrial company that is a world leader in seamless steel tubes sector. One of the primary characteristics of this sector is highly science-based nature of its products. Consequently, significant innovations often cannot be achieved without significant knowledge production. Facing new challenges (such as new regulations and upcoming new standards), the company launched several initiatives following two main traditional approaches: a) *product design* based on an experimental approach based on generating and testing compound materials meeting the standards b) *research projects* to reinforce their knowledge about different aspects (electrochemical, metallurgical, etc.) of the phenomena. Due to the upcoming changes in regulations and standards, timely delivery was a privileged criterion. Another objective of the initiatives was the steady progression in the production of scientific knowledge to prepare the future undertakings.

Our access to the internal processes of the engineering and research departments of the company allowed us to let appear two difficulties. First, sticking to the results of experimental plans decreased the chances for the engineering department to formulate powerful and new product concepts. Second, results obtained by research department could not be re-injected back into the concept development step, especially because, those research were not conducted in connection to any novel product ideas.

Based on C-K theory, we might assume that the problem arises from the inadequate coupling of design and research processes. While this is being investigated and discussed with the industrial, we also modelled and studied the associated strategies in a simple yet controllable way using Graphi.CK. Results show that while pure concept driven (c-driven) strategies achieve better on speed criteria, research based (pure k-driven) strategies allow having more homogenous knowledge production. Moreover, as predicted by C-K design theory, hybrid strategies seem to allow addressing both criteria by increasing robustness without sacrificing too much of speed. It can be argued that this findings stand for better integration between research and design departments. The paper proceeds as follows. In section 2, basic definitions of C-K theory are overviewed. In section 3, the formal model underlying Graphi.CK is presented. In section 4, it is explained how C-K's notions might be implemented using section 3's model. Section 5 presents the experimental setting, the strategies tested and the results. Section 6 present further discussions. We conclude with the limits of the current study.

2. A short overview of C-K design theory

Hatchuel and Weil (2007, 2009) propose a theory of design reasoning which captures some of the fundamental properties of design reasoning process. The theory is based on the distinction and interaction between two spaces:

- A knowledge space represents all the knowledge available to a designer (or to a group of designers) at a given time. These are propositions that the designer is capable of stating as true or false; i.e., propositions whose logical status are known (e.g., some phones are mobile).
- A concept space represents propositions whose logical status are unknown and cannot be determined with a given knowledge space. These are propositions that can be stated as neither true, nor false at the moment of their creation (e.g., some phones prevent heart attacks).

In C-K theory, concepts are descriptions of an object of the form "C: there exist an object x with the properties p1,p2,..., pn" such that C is *undecidable* with respect to K. In other words, the designer who created the concept is not able to tell whether such an object may exist or not. A design process begins if such an undecidable formula can be created. A designer can then elaborate the initial concept C0 by partitioning it - that is, by adding further properties to the C0. In C-K theory, there are two kinds of partitioning. *Restrictive partitions* add to a concept a usual property of the object being designed. *Expansive partitions* (or conceptual expansions) add to a concept novel and unprecedented properties.

Concepts, although different than knowledge in their logical status, they are created from knowledge. For this reason, different designers with different knowledge spaces may create different concepts. A concept space can only be defined with respect to a knowledge space – concepts are *K-relative*.

When elaborating a concept space, a designer might use his or her knowledge, either to partition further the concepts, or to attempt a validation of a given concept (to accept it as true or to reject it as false). Often the validation of a concept is not readily possible. New knowledge warranting the existence conditions of such an object should be acquired. In terms of the theory knowledge should be expanded. The expansion of knowledge space is called K-expansion. The central proposition of C-K theory is thus "design is the interaction and dual expansions of concepts and knowledge".

3. A graph representation as a model of knowledge space

The key notions of C-K theory we have reviewed in the previous section are quite general and they can be interpreted using different formalisms (see e.g. [Hatchuel and Weil 2007], [Kazakci et al. 2008] or [Kazakci 2009]). In Graphi.CK, graph formalism has been used for describing (types of) objects and systems of related objects. The choice of this simple yet general formalism is voluntary since we aim for controllability of the experiment and general phenomena.

3.1 A graph-based formalism to model knowledge structures

We use a graph-based formalism to model knowledge structures. The basic formalism in graph theory is given by a collection **V** of *nodes* and a collection **E** of relations between those nodes called *edges*. Given such formalism, nodes can be associated with objects (or their names). Edges represent a very generic relation between objects; they can be interpreted as "related to".

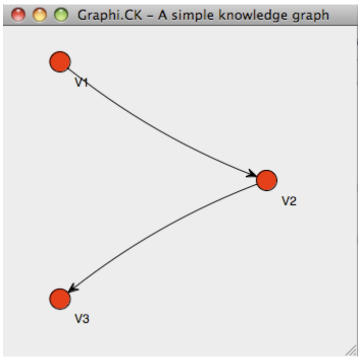


Figure 1. A simple knowledge graph

An example representation is given in figure 1. The node V1 corresponds to an object that is defined with respect to its relation to V2, that is, itself, defined by its relation to V1 and V3. A chain on the graph corresponds to a *sentence* about related objects. For instance the chain V1→ V2→ V3 depicted in figure 1 can be interpreted as the sentence "an object V1 that is related to an object V2 that is related to an object V3". Such (sets of) sentences can be seen as descriptions of sets of related objects within the proposed formalism. This formalism corresponds to a language where atomic propositions are relations (e..g P: V1→V2 and Q:V2→V3). Since no variables are considered, the formalism is propositional. Thus, a sentence in this language corresponds to a *definition* (or description) of a set of objects.

3.2 Underlying Interpretation

A node V in the knowledge graph is interpreted as "the designer knows (or is aware) about the existence of an object V". An edge E between two nodes V1 and V2 is interpreted as "the designer knows that the objects V1 and V2 are related". While existence of an edge between two nodes is interpreted positively as knowing, the absence is not interpreted as negation (i.e. the absence does not mean "*V1* is not related to *V2*"). When a relation exists in K space, it is considered as known (or, true). If this is not the case, it is considered as unknown.

Given a graph G, any of its subgraphs can be interpreted as a description. In particular, when a subgraph is a disconnected component of the graph, it corresponds to a (sub)system of related objects. We refer to such components as *knowledge islands* (or islands, for short). Figure 2 depicts an example knowledge space where appears three islands each of which corresponding to a description of a set of objects related with each other. This can be seen as a simplified model that can capture many empirical situations. For instance, in the case of a material research lab, two disconnected components represent two scientific disciplines that do not share any common object. Cracks experts will build model of cracks which describe how the metal crystalline structure (V1) can be disturbed by some chemical molecules (V2) which provoke cracks (V3) under some pressure conditions (V4). On the other hand metallurgist experts will describe how a tube (V5) is composed of Iron (V6), and some specific elements (for instance C, Mg,…) (V7, V8,…).

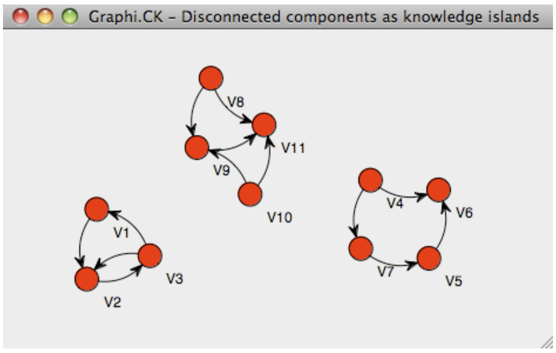


Figure 2. A knowledge space representation

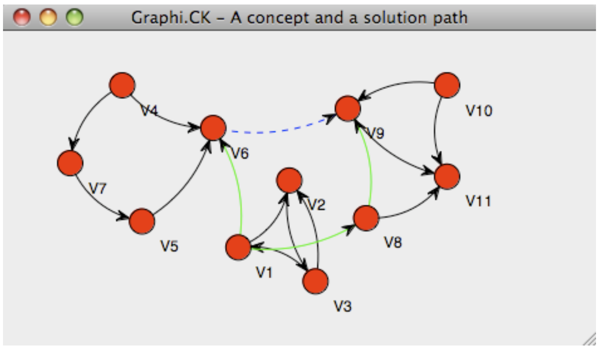


Figure 3. A connected knowledge graph. The chain (V6,V1,V8, V9) is the knowledge expansion that realized the concept (V6,V9)

4. Interpreting C-K notions in the graph representation

4.1 Knowledge expansion

Following our model, elementary learning consists in the addition of a new object or a new relation between two existing objects to the knowledge graph. Under the intended interpretation, if an edge between two nodes does not exist, this corresponds to the unavailability of information. Unavailable information may become available to the designer as a result of learning – new nodes and edges can appear, new connections are established. Furthermore, misinterpretation does not exist. The designer never knows about a relation that does not exist in reality. Hence, learning occurs monotonically and there is no need for revision. Note that this can be seen as an ideal world where eventually everything can become connected (and learnt).

4.2 Concepts, conceptual expansions and partitioning

In C-K theory, a concept is a proposition that is neither true, nor false. In the described formalism, we can interpret C-K definition of concepts as propositions that are not yet known to exist, but which can be learned in future. In our formalism, such propositions represent *voids*, that is, couples (Vi, Vj) of nodes that are not connected by any chain of relations (this interpretation is quite similar with [Kazakci, Hatchuel et al. 2008]). For example, in figure 2, (V6, V9) (dotted blue line) represents such a void. Such propositions can be considered in C space as concepts. Further voids may be added to introduce further expansive partitions, e.g. (V6, V9, V2). If any already existing relation that is already connected to the nodes composing the concept, e.g. V11, is added in C space, this corresponds to a restrictive partition, e.g. (V6,V9,V11).

4.3 Conjunction - the end of a design process

Since a concept can be seen as an attempt to relate previously unconnected systems and types of objects, if a chain of relations connecting one (or more) such systems is found, then, the design has succeeded: a sentence (or a piece of knowledge) relating a new type of system where previously unconnected types of objects are now known to be related has been created.

Said in other terms, given a concept attempting to connect two knowledge islands, if K-expansions occurs in such a way that a chain passing through the nodes forming the concept's definition can be found, then these nodes have been learnt to be in relation with each other. Comparing figure 2 and 3, we see in figure 3, an example of such a chain connecting previously unconnected subgraphs: (V6,V9) is no longer a concept since the designer *knows* that these nodes are now related by (V1→V6, V1→V8, V8→V9). Such a knowledge expansion might be considered to model the first working prototype that confirms to the designer the feasibility the initial concept. Any further connection (between the previously unconnected components) that the designer might learn over the time, only confirms that parts of such systems are related. Although an increase in the connectivity of the knowledge islands (e.g. further cycles between the components) might be interpreted as a better comprehension by the designer of the system's nature and relationships between its parts.

5. Simulating c-driven and k-driven explorations

To illustrate the approach and the kind of results we can access, we present here some example experiments contrasting and comparing two exploratory strategies adopted by the industrial company (presented in the introduction), namely, concept-driven reasoning and knowledge-driven reasoning. Those two strategies correspond to the two kinds of orientations adopted by traditional engineering design and research departments involved in such projects. The pseudo-code for the major algorithms used for the current experiments are given in Table 1.

5.1 Generic setting of the experiments: detecting and exploiting voids

In our implementation, an initial knowledge space structure is being progressively generated by an algorithm based on three parameters:

- *ndc*, the number of disconnected components
- *nn*, the number of nodes in disconnected components
- *c*, the connectivity of each node within its component

In a given experimentation, *ndc* is kept constant whereas *nn* and *c* may be made to vary randomly (for each disconnected component) depending on the experiment setting. Since this graph is constructed by successive addition of disconnected components, it becomes possible to maintain a hashmap relating each node to its component id. As a result of this, the system has at any moment a kind of meta-knowledge on *what it does not know* – the *voids* that form the undecidable propositions.

Based on this setting, once a knowledge structure has been generated, knowledge expansion or concept partitioning can be implemented by generation of candidate voids and a selection among them. In our experiments, we used simply a random selection.

Currently, all the alternative expansions are generated for consideration. Note that this models a rather extreme case: with *ndc* and *nn* increasing, generating all possible combinations becomes a heavy task. Once the alternatives are generated, a random selection is made and the proposition is added: i) in C space, if the system is looking to create a concept; ii) in the K space, if it is looking for a k-expansion.

5.1.1 Concept driven reasoning

Intuitively, Concept-driven Reasoning (CDR) needs to use the information encoded by the concept description. In the case of a single partition (e.g. the initial concept) this information is simply the name of the two nodes that we are looking to connect. For instance, if the initial concept tries to connect V1 and V2, the only information that we have in order to guide the exploration and expansion in K space is V1 and V2. This models the idea that if we are trying to design a *flying ship*, we start out by inquiring our knowledge base about what is a *ship* and what does it mean to *fly*.

In C-K theory, there are two ways such an inquiry may proceed: First, some knowledge associated with the inquiry term will be activated (or remembered). For instance, a *ship is an object floating on the water.* Second, a k-expansion may occur *floating in the air is flying* (assuming this was not known by the designer previously). In our framework, the first case may be modeled simply by the retrieval of an associated piece of knowledge by the adjacency information contained in the graph (e.g. the ship node is connected to floating node). The second case is modeled by the addition of a new edge in K space between two unconnected nodes.
Hence, the exploration and the expansion of K space triggered by a (single) partition of C space can be seen as a progressive activation of nodes, recruiting connected ones and connecting new ones at each time step. Internally, this corresponds to a list of nodes starting with the two nodes defining the concept. Every time an expansion (or an activation) is operated, the node that is recruited is added to this list as a candidate from which further expansions (or activations) can be made in the next steps. In the current experiments, we only considered expansions. In the present experiments, only k-expansions are considered.

Table 1. Pseudo-codes for major algorithms used in the present experiments

Pseudo-code for Void selection if number of k-islands greater than 1 Select a random k-island p1 from list of k-islands Select a random k-island p2 from list of k-islands repeat previous step while p1=p2. Select a node V1 (randomly) from p1's nodes Select a node V2 (randomly) from p2's nodes end if return V1, V2	Pseudo-code for K-driven learning Select a concept (select void V1,V2 and Add V1-V2 in C space) while k-island of V1 different than k-island of V2 Select void V3,V4 and Add V3-V4 in K space (K-expansion) end while return
Pseudo-code for C-driven learning Declare List: path (path contains nodes from which k-expansions are made) Select a concept (select void V1,V2 and Add V1-V2 on C space) Add V1 and V2 to path while k-island of V1 different than k-island of V2 Select random element n-temp from path Select a node V3 which does not belong to n-temp's k-island k-expand: add n-temp-V3 to K space add V3 to path end while return	Pseudo-core for Hybrid learning Declare List: path (path contains nodes from which k-expansions are made) Select a concept (select void V1,V2 and Add V1-V2 on C space) while k-island of V1 different than k-island of V2 Alternate between 1) and 2) at each "while" iteration 1) Select void V3,V4 and Add V3-V4 in K space (K-expansion) 2) Select random element n-temp from path Select a node V5 which does not belong to n-temp's k-island k-expand: add n-temp-V5 to K space add V5 to path end while return

5.1.2 K- driven strategy

By contrast to concept-driven learning, it is possible to conceive a knowledge-driven strategy (KDR) where new information is acquired independently of any concept. In the general case, the objective is to learn as much as possible about each and every node in the graph. This can be interpreted as a *densification of the knowledge base* in order to increase its connectivity and reduce the number of disconnected components. Note that if the learning occurs only between disconnected components, new paths that connect previously unconnected nodes appear. This kind of learning models a very particular strategy since cross-domain knowledge production is privileged. Traditionally, a research laboratory would most of the time restrict its knowledge production to its base discipline.

5.1.3 Hybrid strategy

We have formulated a third and final strategy that we call hybrid reasoning. In the industrial setting we have previously described, such strategies correspond to an effort for *coupling* engineering design and research efforts so that the effects and interactions are reinforced. In practical settings, a multitude of hybrid models can be conceived. Here, we use a straightforward combination of the previous two strategies: at each learning step, the algorithm realizes alternatively either a learning driven by the concept or a learning driven by knowledge. This general way of modeling interactions between CDR and KDR allows observing the effects a basic hybrid strategy rather than any conceivable sophistication or additional heuristics.

5.2 Criteria for design performance: speed vs. connectivity

It is possible to use at least two contrasting criteria for measuring the impact of the previously described exploration strategies, namely, the speed of conjunction and the connectivity of the resulting knowledge base. These two criteria fit indeed some of the primary concerns expressed by the industrial partner. We chose the following indicators for these criteria.

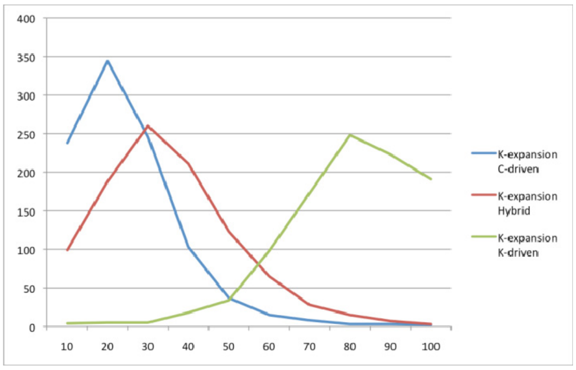
1. Speed of conjunction: number of expansions needed to find a solution. It corresponds to the classic industrial goal to reach in a limited time span at least one marketable solution of the new material
2. Connectivity: increasing connectivity corresponds to a densification of knowledge base. A more connected graph is interpreted as better understanding of the considered phenomena. A common measure of connectivity in graph theory is entropy-based metrics. We use the following metric:

$$E(g) = -\frac{\sum_{V \in \mathbf{V}} p(V)\log(p(V))}{\log(|\mathbf{V}|)}$$

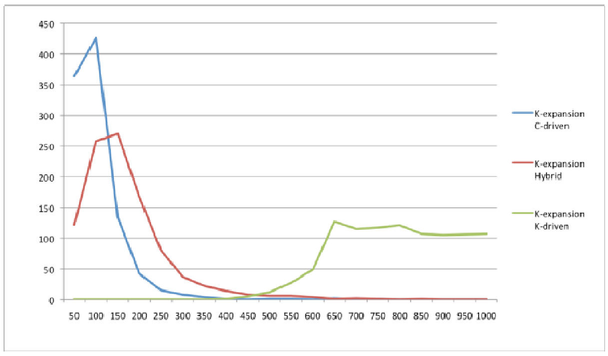
where $p(V) = ind(V)/2|E|$, $ind(V)$ is the number of edges incoming to the node V, $|\mathbf{V}|$ is the total number of nodes, $|E|$ is the number of edges. With this metric, the more each node is connected, the more is the entropy. When connectivity reach maximum, entropy becomes maximum.
The connectivity of a knowledge space can be seen as an indicator of *robustness*. In industrial terms, several types of robustness can be conceived. One possible meaning is that equivalent satisfying solutions can be found through different designs allowing for industrial flexibility (various processes are possible). It also provides backup solutions in case unexpected problems arise. Finally, it also means that there is a better scientific understanding of how and why existing designs really work. While entropy represents (directly or indirectly) all of these dimensions, other measures can be introduced as well. Those will be presented in subsequent work.

5.3 Experiments and some results

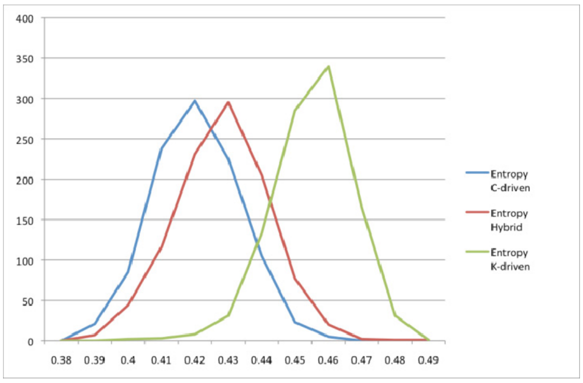
Due to the space restrictions we present here only the result of single run experiments. Single run experiments correspond to a single run of a given strategy using different topology of graphs. We have realized the experiments for all three strategies with graphs of varying topologies ($ndc \in [100..100]$, $nn \in [2..10]$, $c \in [1..nn\text{-}1]$). 1000 thousand run have been realized for each strategy. For each strategy, values obtained for each of the suggested evaluation criteria have been used to build distributions (instead of using e.g. mean values). Characteristic examples of these distributions are given in figure 4. These experiments have been repeated several times for each experimental configuration to check for stability of results. Indeed, we have observed a solid invariance for the observed patterns and stability of distributions beyond ~100 iterations. Furthermore, the size of the islands and their connectivity seems not to matter. This might be expected since the described strategies do not exploit intra-island information. The graphs (a) and (b) in figure 4 represents the distribution of number of k-expansions (the number of edges added to K) needed to have a conjunction over a sample of thousand runs of the simulation. For instance, on a thousand runs with 100 islands, the number of k-expansion needed for concept-driven learning were between [0,10] for slightly less than 250 times and between (10,20] were about 350 times, etc.
Looking at the general medians of (a) and (b) distributions, we can see that the concept-driven reasoning needs (in general) much less learning in order to have a conjunction than k-driven reasoning. Also, when the number of islands increases from 100 to 1000, c-driven learning strategy preserves its performance. %80 of conjunctions occur before 1/3 of the islands have been connected.



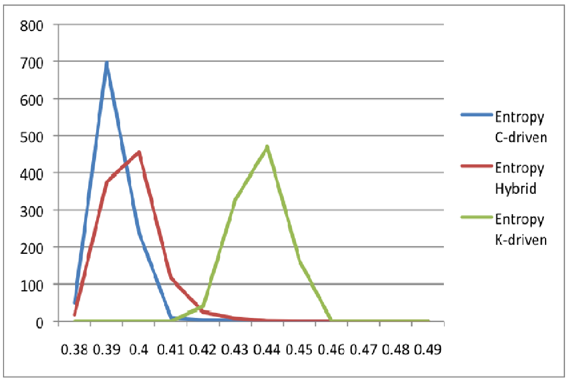
(a) K-expansion for 100 islands



(b) K-expansion for 1000 islands



(c) Entropy for 100 islands



(d) Entropy for 1000 islands

Figure 4. Probability distributions of Speed and Robustness (measured respectively by the number of K-expansions necessary for conjunction and Entropy of the resulting graph)

When comparing k-driven strategy, the only notable difference between 100 and 1000 islands is the flatness of the distribution in (b). When the number of knowledge islands increases significantly (from 100 to 1000), the probability of having a conjunction with a k-driven strategy before half of the islands are connected is nearly 0. Within the model, this means that although a significant amount of cross-domain knowledge production could be made, in product design terms no single innovation has been made. Moreover, the flatness of the distribution suggests further imprecision on when the process might terminate.

The graphs (c) and (d) represent the distributions of final entropy of the K space for different strategies. Starting with same K space structure, we observe that, the entropy induced by k-driven learning is in average significantly superior to c-driven and hybrid strategies. The main reason for the difference is due to the fact that K-driven strategies build the knowledge base evenly along several dimensions. Said in other terms, different islands of knowledge are given (more or less) equal importance and priority for organizing the learning process. Not only the resulting graph with k-driven is more connected, it is also more evenly dispersed.

A somewhat surprising finding is that in both cases and for both criteria, the hybrid learning that may be expected in-between the two strategies, appears much closer to c-driven. We can observe that the hybrid strategy competes decently with c-driven in early conjunctions. In (a), while c-driven has its pike around 20 expansions, hybrid has his around 30. Then, its probability of conjunction decreases steadily following c-driven closely. A similar pattern occurs for 1000 islands. As to the connectivity, hybrid strategies perform better than C-driven, though without really competing K-driven strategies. Although, hybrid strategy alternates evenly between C-driven and K-driven strategies, the effect of C-driven reasoning for transforming K space seems more powerful (both in terms of speed and entropy).

Overall, we can state that a hybrid strategy where the intensity of K-driven learning is well modulated with respect to the deadlines of the design project would allow the simultaneous optimization of both criteria.

6. Discussion

6.1 Interpreting the efficiency of hybrid strategies: the partitioning power of out-of the box knowledge

In the light of the previous results and given the framework of our model, we can observe that C-driven strategies are fast but do not promote a deeper understanding of the phenomena. Since the production of new knowledge is limited, a profound revision of the initial product concept is not possible. On the other hand, K-driven strategies allow establishing cross-domain links and a variety of new connections that might be introduced into the concept appear. However, due to its slow speed it may be of limited use for projects that operate under tight schedules. Hybrid strategies allowing a more balanced exploration seem to address both criteria.

What happens in hybrid strategies is a phenomenon that can be labeled as the *partitioning power of out-of-the-box knowledge*. In such strategies some effort is given to increase understanding (connectivity) through new knowledge. However, such new knowledge becomes immediately potential alternative paths for C-driven strategies. If the new knowledge offers no such alternatives, the former c-driven strategies are unchanged but robustness is not increased either. If it offers alternative paths c-driven solutions will be more varied and robust. This finding highlights the conditions for out-of the box strategies in engineering design. Radically new concepts need new knowledge to be expanded into valid designs. Yet, *creating new knowledge through research could take a long time before an alternative design will come up*. Thus as C-K theory predicts robust innovative design will appear when the new knowledge is combined with a concept-driven learning by hybrid strategies. These insights from the simulation study contributed to a new approach developed with our industrial partner. On the one hand, a systematic concept-driven tree structured experimental strategy was built using existing knowledge; on the other hand, a list of research projects was established and the "partitioning power" of each of this project was evaluated on the basis of the established concept tree. Currently the two programs are running and their combination has already reached both a small set of solutions and a specific set of research questions that could directly improve them. The details will be presented elsewhere.

6.2 On the difficulty of detecting voids – preparing strong expansions

Beside the results of experiments, some of the features of the model and its potential to capture real life design reasoning can be insightful to discuss. One such feature would be the detection of voids that are used both to create concepts and to guide the learning process. For a given graph, detecting voids require the traversal of the graph, e.g. by breadth-first search, for detecting disconnected subgraphs. This proceeds by selecting a node and checking its neighbours, the neighbours of its neighbours and so on. All the nodes that cannot be reached this way necessarily belong to a different and disconnected component of the graph. A relation from any node in the starting component to any node in the set of unreached nodes of the graph can be seen as a concept (in our model, they are all relations that do not belong to the graph). Note that the generation of *all* the concepts with respect to any given state of the graph requires several traversal of the graph by breadth-first search algorithms; once a second disconnected component has been found, the first one is eliminated and the algorithm is rerun from a node of that second component. If there are still some unreachable components, then, the process is repeated (until no such component remains). Since the worst-case performance of breadth-first search's complexity is $o(|V|+|E|)$ (every node is connected to every other node and all the paths passing from all the nodes are visited), this operation is quite costly. This theoretical difficulty matches well real life design efforts where what is known need to be regenerated in order to detect what is not known – the voids that lead to concepts.

7. Concluding remarks: Limits and further development

The paper presented Graphi.CK, a simulation model based on C-K theory, and an example application with some initial results. We have tested and compared two basic design strategies, C-driven and K-driven, inspired from the case of a partner industrial company. We have found that while C-driven strategies were faster, K-driven strategies provided a more robust learning process. It is also demonstrated that Hybrid strategies allowed increasing connectivity of the knowledge base without sacrificing too much speed. As remarked in the introduction, interaction with a simulation model is a powerful method for synthesizing new hypothesis and concepts concerning the reality that is being modeled. As presented, our interaction with Graphi.CK so far has allowed us to think about hybrid strategies and about a notion of “partitioning power” of new knowledge. Both notions are being operationalized in the context of the previously described industrial setting.

Graphi.CK is still a nascent project. Several limitations need to be addressed in future work for further development:

1. Limits inherent to the formalism: The formalism used in Graphi.CK is quite basic. Many enhancements can be imagined. However, for the time being, we prefer keeping it simple yet controllable in order to ensure that we have explored to the widest possible extent some basic phenomena.
2. Limits of the currently considered design strategies: The design strategies that have been explored in this paper are also basic. We can especially note that they do not take into account an interactive evolution of C and K spaces and that we only used a single expansive partition as concepts. A wide range of sophistication can be conceived. We are currently building a repertory of different strategies based on related literature and our past and current experiences with industrial companies – some of which are already being tested.
3. Limits of the present results: The presented results can be (and in fact, have been) investigated more in depth. Currently, several other results are available and these are being compiled for an extension of the present work.

Acknowledgement

This research has been supported by *Design Theory and Methods for Innovation* Chair of Mines ParisTech. We would like to thank our industrial partner for valuable discussions.

References

Hatchuel, A. and B. Weil (2007). Design as Forcing:Deepening the foundations of Ck theory. 16th International Conference on Engineering Design - ICED 2007, Knowledge, Innovation and Sustainability, Paris, France.

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

Kazakci, A. (2009). A formalisation of CK design theory based on Intuitionist Logic. International Conference on Research into Design. ICORD09. A. Chakrabarti. Banglore, India, Research Publising Services: 499-507.

Kazakci, A., A. Hatchuel, et al. (2008). A model of C-K design theory based on a term logic: a formal background for a class of design assistants. International Design Conference 2008, Dubrovnik, Croatia.

Le Masson, P., B. Weil, et al. (2006). Les processus d'innovation- Conception innovante et croissance des entreprises Paris, Hermès.

Saunders, R. and J. S. Gero (2002). Curious agents and situated design evaluations. Agents in design. J. S. Gero and F. Brazier, Key Centre of Design Computing and Cognition, University of Sydney, Australia: 133-149.

Sosa, R. and J. Gero (2003). "Social change: Exploring design influence." Lecture notes in computer science ISSN 0302-9743 2927: 106-119.

Sosa, R. and J. Gero (2003). Design and change: A model of situated creativity. Approaches to Creativity in Artificial Intelligence and Cognitive Science, IJCAI03, Acapulco

Taura., T. and Nagai. Y. (2008). Design Insight-A Key For Studying Design Creativity.T. NSF International Workshop on Studying Design Creativity' 08. AIx-en-Provence, France.

Wolfram, S. (2002). A new kind of science, Wolfram Media Inc.

Dr. Akin Osman Kazacki
CGS - Centre de Gestion Scientifique -Mines ParisTech
60 boulevard Saint-Michel, 75272 PARIS Cedex 06
Telephone: 00 33 1 40 51 92 08
Email:akin.kazakci@ensmp.fr