INSTITUTO TECNOLOGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS MONTERREY

DIVISION DE INGENIERIA Y ARQUITECTURA

PROGRAMA DE GRADUADOS EN INGENIERIA

TECNOLÓGICO DE MONTERREY

PRINCIPLES TO AUTOMATE INVENTIVE PROBLEM SOLVING BASED ON DIALECTICAL NEGATION, ASSISTED BY EVOLUTIONARY ALGORITHMS AND TRIZ

TESIS

PRESENTADA COMO REQUISITO PARCIAL PARA OBTENER EL GRADO ACADEMICO DE:

DOCTOR EN CIENCIAS DE LA INGENIERIA

POR:

ROBERTO ALEJANDRO DURAN NOVOA

MONTERREY, N. L. MAYO DE 2011

INSTITUTO TECNOLOGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS MONTERREY

DIVISION DE INGENIERIA Y ARQUITECTURA

PROGRAMA DE GRADUADOS EN INGENIERIA

TECNOLÓGICO DE MONTERREY

PRINCIPLES TO AUTOMATE INVENTIVE PROBLEM SOLVING BASED ON DIALECTICAL NEGATION, ASSISTED BY EVOLUTIONARY ALGORITHMS AND TRIZ

TESIS

PRESENTADA COMO REQUISITO PARCIAL PARA OBTENER EL GRADO ACADEMICO DE:

DOCTOR EN CIENCIAS DE LA INGENIERIA

POR:

ROBERTO ALEJANDRO DURAN NOVOA

MONTERREY, N. L. MAYO DE 2011

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS
SUPERIORES DE MONTERREY
CAMPUS MONTERREY
DIVISIÓN DE INGENIERÍA Y ARQUITECTURA
PROGRAMA DE GRADUADOS EN INGENIERÍA

TECNOLÓGICO DE MONTERREY®

PRINCIPLES TO AUTOMATE INVENTIVE PROBLEM SOLVING BASED ON DIALECTICAL NEGATION, ASSISTED BY EVOLUTIONARY ALGORITHMS AND TRIZ

TESIS
PRESENTADA COMO REQUISITO PARCIAL
PARA OBTENER EL GRADO DE
DOCTOR EN CIENCIAS DE LA INGENIERÍA

POR:

ROBERTO ALEJANDRO DURAN NOVOA

MONTERREY, N.L. MAYO 2011

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY
CAMPUS MONTERREY

ENGINEERING DIVISION
ENGINEERING GRADUATE PROGRAM

Committee members recommend that the research project presented by Roberto Alejandro Durán Novoa be accepted as a partial requirement to obtain the academic degree:

Doctor in Engineering

APROVED BY:

Dr. Noel León Rovira
Chair of the committee

Dr. Humberto Aguallo Téllez
Committee member

Dr. David Güemes Castorena
Committee member

Dr. Víctor Hiram Vázquez Lasso
Committee member

Dr. Eduardo Uresti Charre
Committee member

Dr. Alex Elías Zúñiga
Director
Doctorado en Ciencias de la Ingeniería

May, 2011.

En el medio de un bosque un árbol cae. Nadie lo vé, nadie lo escucha, sin embargo cae.

Gracias, Rosario y Belina.

Acknowledgements

It is a pleasure to thank those who made this thesis possible...

... Dr. Pedro Sariego Pastén, for encouraging me to study the Doctorate and contacting me with the ITESM

... Dr. Noel León Rovira, for his constant support, patience, and feedback during all this time

... Dr. Roland De Guio, for allowing me to participate in his laboratory, expanding my design vision

... Dr. Humberto Aguallo Téllez, Dr. David Güemes Castorena, Dr. Víctor Hiram Vázquez Lasso, and Dr. Eduardo Uresti Charre, for giving me their valuable time and knowledge by accepting to be members of my doctoral committee

... Dr. Gaetano Cascini, for giving me his valuable time and counsel during the last stage of my thesis

... Dr (c) Jorge Lozoya, for his friendship and constant inspiration

Acronyms

40-IP: 40 inventive principles

CS: computer sciences

C-K: Concept- Knowledge

CAI: Computer aided innovation

DiP: Dialectical principles

DNA: Dialectical Negation Algorithm

DOE: Design of experiments

EAs: Evolutionary Algorithms

GAs: Genetic Algorithms

GP: Genetic Programming

IP: Inventive Problem(s)

IPS: Inventive Problem Solving

NoP: Network of Problems

PDP: Product development process

QFD: Quality function deployment

SIT : Structured Inventive Thinking

TRIZ: Theory of Inventive Problem Solving

USIT: Unified Structured Inventive Thinking

Abstract

An inventive problem (IP) can be defined as a human perception of a situation that has to be changed, but with at least one obstacle which impedes achievement of the desired goal. In practice, they are solved generally using random trial and error, despite the fact that in literature there are several structured approaches to stimulate creativity and deal with them. During the development of products IP solving is particularly important throughout the concept generation, being often conducted intuitively by field experts. This dependence of intuition and expertise is a bottleneck in the design as a whole: intuition for being unpredictable, and expertise for being rare and thus expensive.

This dissertation proposes a series of steps, based on dialectics, to decrease IP solving user dependence. First, the reasons behind this dependence are investigated, describing the critical tasks to be performed and developing the necessary characteristics of the tools to be utilized, simplifying several existing ones under a coherent framework that can withstand different levels of expertise.

In order to explore and develop the proposed model, it is studied the complementation between people's innate creativity and Computer Sciences, aiming to ideally solve IP automatically. Due to its empirical results, Evolutionary computing and TRIZ techniques are utilized for the development of the study cases, whose results shows the viability of the dialectical hypothesis in the concept generation. Finally new and complementary research directions, likely to deliver concrete results, are proposed.

Keywords: Invention, optimization, dialectics, Triz, evolutionary algorithms

Table of Contents

1. Introduction to research

Everyday there are problems that need to be solved. A trivial one has a solution that is part of the body of knowledge and consequently does not presents additional obstacles to achieve the desired result, reason why they are often called "routine problems" or "tasks", in order to differentiate them from the inventive problems (IP). An IP can be defined as a human perception of a situation that has to be changed, but with at least one obstacle which impedes achievement of the desired goal. In practice, IP are solved generally using random trial and error, despite the fact that in literature there are several structured approaches to stimulate creativity and deal with them (Heuristics, Mind Mapping, Brain Storming, Morphological Matrix, C-Sketch, TRIZ, others (Ulrich & Eppinger 2003; Polya 1988; Otto & Wood 2001).

During the development of products this situation is particularly important at the design stage, mostly because of the "fuzzy front-end": the phase between first consideration of an opportunity and when it is judged ready to enter the structured development process. Considered the greatest weakness in product innovation, managers declare familiar symptoms when it fails (Khurana & Rosenthal 1997):

- New products are cancelled in midstream for not matching the "company strategy"
- Key people are too busy to focus in top priority projects.
- New products frequently are introduced later than announced because the concept has become a moving target.

In words of Khurana, "the failure to integrate a product strategy, a well-planned portfolio, and a facilitating organization structure with clearly identified customer needs, a well defined product concept, and a project plan can severely hamper new product development" (see Figure 1) The previous can be resumed in that the lack of control blocks innovation, and that depends mostly on the concept development stage, which also happens to determine most of the product cost (Figure 2). Consequently, any improvement regarding concept generation is likely to have a significant impact in product development and problem solving in general.

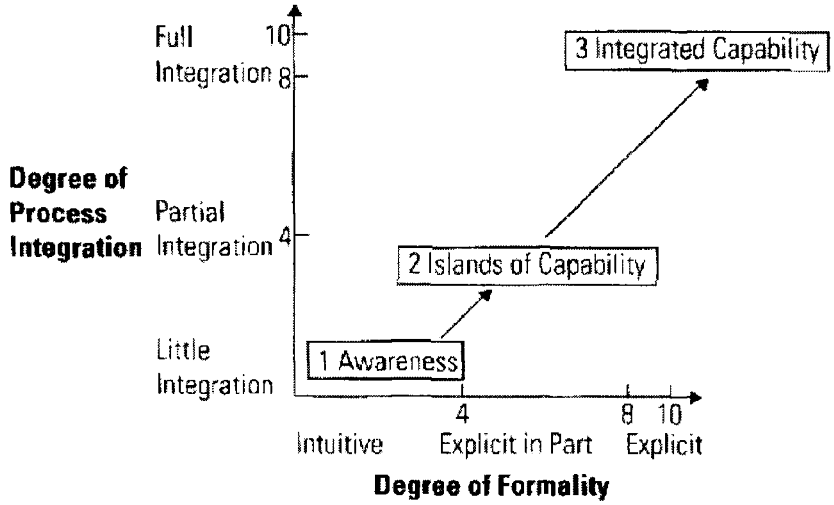

Figure 1: Stages in the transition to a mature Front End, fundamental to have a reliable innovative process. (Khurana & Rosenthal 1997)

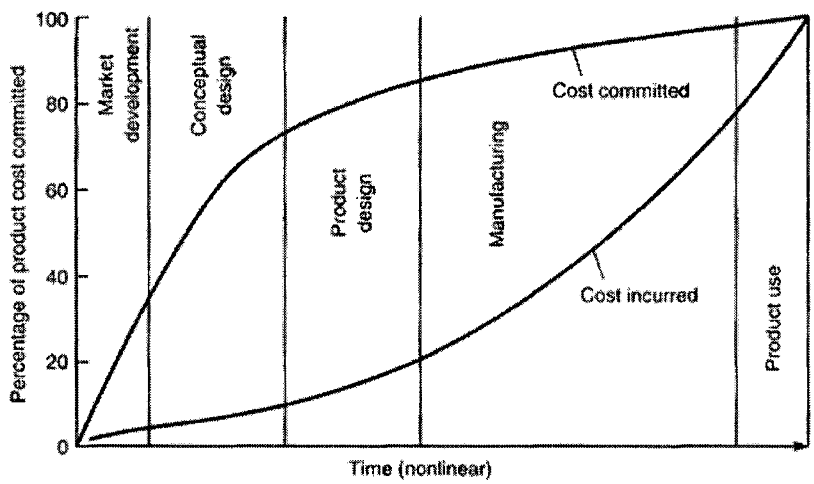

Figure 2: Product/Cost commitment during phases of design (Dieter & Schmidt 2009)

Defining concept as "an idea that is sufficiently developed that it can be evaluated in terms of physical realizability" (Dieter 1999), it can be said that the search of concepts is conducted intuitively by field experts until the moment of evaluation. This dependence of intuition and expertise[1] is a bottleneck in the design as a whole: intuition for being unpredictable, and expertise for being rare and thus expensive.

[1] Definitions for intuition, expertise, concept, etc, are included in the Glossary.

Due to the previous, this dissertation is focused in Inventive Problem Solving (IPS) during the concept generation stage of design, aiming to contribute to make their solution process more controllable and consequently closer to be automated. This augmented control would open the possibility to utilize computers' capacities of performing repetitive task to actively aid the concept generation, in a similar manner that optimization is perform nowadays. As a consequence the computer sciences (CS) techniques related to the creative process are another relevant area to consider, standing out Evolutionary Algorithms (EAs) due to its results solving practical problems (section 2.2)

To deal correctly with this problematic, it is first necessary to clarify the reasons behind the dependence of intuition and expertise and the IPS methods current state of the art. This will tell about the conditions that a system must deal with to develop valid IPS concepts, determining what exists and what is missing, delivering the chance of making legitimate and useful proposals. The development and validation of those proposals constitutes the logical next step.

The present dissertation starts with the formal declaration of the problematic, (chapter 1), followed by the literature review (chapter 2), and the development of a model proposal (chapter 3), which is explored with the study cases (chapter 4). Finally, the analysis and conclusions (chapters 5 and 6) summarizes the work performed and results, indicating future development directions.

1.1.Problem Description

As mentioned at the beginning of the chapter, the IPS process, especially the concept generation during the product and development process, is highly dependent of intuition and expertise. This has the effect that "the heart of the product development process" (Dieter & Schmidt 2009) can be turn into an uncontrolled process that relies on human inspiration. Several researchers have studied IPS, and there are numerous problem solving theories and methods that have shown usefulness in different contexts. However, it was confirmed by the literature review (chapter 2) the perceived idea that most of the methods have a limited range of application, do not deal with fundamentals of IPS, and often give to the users more tools that the ones they can handle. These situations are presented and analyzed in sections 3.3.1 and 3.3.2.

Considering the definitions of IP and designers, as everyone "who devises courses of action aimed at changing existing situations into preferred ones" (Simon 1996), this dissertation proposes a series of steps to solve IP based on dialectical negation, by describing the critical tasks to be performed and the necessary characteristics of the tools to be utilized. Those characteristics are the basis for selecting EAs and TRIZ methods for the development of study cases, whose results will support or not the IPS model proposed, giving direction to the concept generation utilizing CS.

1.2.Research Question

The problematic can be resumed in the following question: Why IPS is so dependant of intuition and expertise, and how to reduce this dependence? The second part of the question can be rephrased:

- Is it possible to generate a practical model to increase IPS independence about the user?

The previous, is the reason why focusing in IPS during the concept generation stage of product development, which is generally considered composed of two stages of (Pahl et al. 1996):

- The *task clarification*
- The *conceptual design*

The problem presented is admittedly too general to be solved in one dissertation, so the research will be focused following the next sub-questions.

1.2.1. Sub-question 1

Which are the problems when solving problems? This question is fundamental, since it gives direction and meaning to the different proposals. If some work does not help to solve the identified difficulties in problem solving, then is not valuable for this research in particular.

1.2.2. Sub-question 2

How actual methods are dealing with those problems, including the user? For example, a problem that requires specific knowledge is not going to be solved by conceptual analysis, and complementary, a large database is not necessarily useful in a problem that requires a qualitative development (new concept generation).

Since a concept must be evaluated physically, this stage needs to be formalized, minimizing the ambiguity existing in many creative methods, more related to psychology ,than to product development (like De Bono, Glover, Osborn).

If the methods are not evaluated, the probabilities of focusing wrong the efforts since the beginning increases.

1.2.3. Sub-question 3

How to make a model as user independent as possible, increasing the automation of IPS in order to utilize CS to actively aid the concept generation? This implies to clarify the considerations to do and effective and efficient transition from a human context into a computational one, considering the way to deal with object complexity.

At the present (2011) IPS, especially concept development is an area minimally explored using CS, being typically related to optimization paradigms, not inventive ones. To develop computational tools could lead to obtain faster results, but more importantly, the development of an algorithmic method able to deliver concrete results is a validation of the proposed model itself.

1.3.General Hypothesis

The main theory utilized to guide the answer search is *dialectics*. Since dialectics explains how the changes operate, it is believed that its principles can be used to model IPS and, based on them, to increase the automation of the concept development. In other words, the main hypothesis is that "dialectics" is an adequate framework to develop a more user independent IPS method.

In a more specific level, TRIZ and EAs have shown results that support the idea that inventiveness can be understood and developed systematically. It is believed that their evolutionary spirit (technical systems in TRIZ and biological systems in EA's) will make natural the emergence of similarities and synergies between them and dialectics, allowing their redefinition to support the solution of problems beyond evolutionary optimization, trespassing Pareto fronts when needed and consequently increasing the automation of concept generation.

In a simplified way, dialectics can be the framework to understand and cross the quantitative-qualitative barrier in IPS (Pareto front), and the EAs the method to do it in an automated manner, using TRIZ principles as a way to reduce the search space to make viable the process as a whole.

1.4.Research Objectives

From a research point of view, this dissertation aims to be a contribution to solve the mentioned problems, simplifying several existing tools under a coherent framework that can withstand different levels of expertise, reducing the concept generation dependence on the user, increasing its general predictability and allowing a more natural flow within the different IPS methods and contexts.

The practical objective is to set the bases to start the generation of computer IPS software, aiming to (ideally) solve inventive problems automatically, increasing the complementation between people's innate creativity and the concepts of Computer Aided Innovation (CAI).

1.5.Relevance and benefits of this research

In general, this research will contribute to diminish the IPS method dependence of the user, having the effect of increasing the predictability and decrease the cost of concept development. More specifically, it will allow:

- Understanding the roots of IP difficulties.
- Give direction to focus future research proposals, pointing towards non redundant research.
- To have a more comprehensive model able to extend the actual application of IP solving methods.
- To develop practical principles for each particular context, derived from the findings.

1.6.Delimitations

It is not the objective of this dissertation to discredit or to replace the tools mentioned in the literature review. Each one is part of the state of the art for its own merit.

The research is focused in the concept generation stage of IPS, not the whole chain of product development.

Is not the intention to provide "the" model to solve IP, but to prove that the IPS model proposed explains better the mechanism of invention than the current state of the art, and that consequently it can deliver solutions with less dependence of intuition and expertise. This implies to reduce (human) trial & error.

Computer programming has been visualized as the way to increase predictability and decrease cost, but the objective of the dissertation is not to develop software, but to solve IP.

1.7.General roadmap

The chapters of the dissertation are:

1) Introduction to research: it presents the formal declaration of the problematic, research question, hypothesis, objectives and delimitations
2) Review of the literature: it presents the most relevant literature regarding inventive problem solving and the principles of evolutionary computing. This material defines the model proposal.
3) Dialectic Model: the analysis of the problematic and the model proposal to deal with it.
4) Study cases: validation of the previous ideas (proof of concept).
5) Obtained results and analysis
6) Research conclusions: critically summarizes the work performed and the obtained results, indicating future development directions.

Complementing all the previous, at the end of the document are included the bibliography, the list of tables, the list of figures, and the glossary.

2. Review of the literature

Within IPS methods there are several points in common, but also deep differences about their aims, scope, efficiency and effectiveness. On this chapter, there are briefly presented and analyzed several IPS methods and the most relevant literature connecting them and EAs, in order to find convergences and missing spots. Eight of those methods were selected because of its popularity[2] IP (Heuristics, Brainstorming, Synectics, Conceptual decomposition, Morphological analysis, Axiomatic design, C-K theory, TRIZ) and two because of its declared characteristics (Concept Generator, SIT), that fitted the practical kind of analysis needed at a particular moment of the present research (to be presented in section 3).

2.1. Creativity related design methodologies

The central point of this literature review is to verify how the selected methods deal with the most generic activities of problem solving[3]:

- Define the problematic
- Create solving concepts
- Develop a proposal
- Evaluate proposal's performance

2.1.1. Heuristics

In his book *How to Solve It* George Pólya provides general *heuristics* for solving problems of all kinds as a four-step approach (Polya 1988). The first distinction is to separate the two main parts of problem solution: the systematic deductive and the experimental inductive parts.

The four mayor steps in problem solving will be:

a) Understand the problem: unknowns? Available data? Circumstances and restrictions?
b) Conceive a plan that links the unknown with the data
c) Execute the plan
d) Analyze the obtained solution

---

[2] Methods usually mentioned in literature, like (Dieter 1999; Altshuller 1999; Otto & Wood 2001), except for C-K theory, which has increased its popularity since 2002.

[3] The activities are the typical mentioned (with slight variations) in the design literature.

2. Review of the literature

Within IPS methods there are several points in common, but also deep differences about their aims, scope, efficiency and effectiveness. On this chapter, there are briefly presented and analyzed several IPS methods and the most relevant literature connecting them and EAs, in order to find convergences and missing spots. Eight of those methods were selected because of its popularity[2] IP (Heuristics, Brainstorming, Synectics, Conceptual decomposition, Morphological analysis, Axiomatic design, C-K theory, TRIZ) and two because of its declared characteristics (Concept Generator, SIT), that fitted the practical kind of analysis needed at a particular moment of the present research (to be presented in section 3).

2.1. Creativity related design methodologies

The central point of this literature review is to verify how the selected methods deal with the most generic activities of problem solving[3]:

- Define the problematic
- Create solving concepts
- Develop a proposal
- Evaluate proposal's performance

2.1.1. Heuristics

In his book *How to Solve It* George Pólya provides general *heuristics* for solving problems of all kinds as a four-step approach (Polya 1988). The first distinction is to separate the two main parts of problem solution: the systematic deductive and the experimental inductive parts.

The four mayor steps in problem solving will be:

a) Understand the problem: unknowns? Available data? Circumstances and restrictions?
b) Conceive a plan that links the unknown with the data
c) Execute the plan
d) Analyze the obtained solution

[2] Methods usually mentioned in literature, like (Dieter 1999; Altshuller 1999; Otto & Wood 2001), except for C-K theory, which has increased its popularity since 2002.

[3] The activities are the typical mentioned (with slight variations) in the design literature.

Like many methods, heuristics tells *what* it has to be done, but not *how* to do it. It is almost impossible to disagree about the 4 steps proposed, but they are so general that its practical utility is minimal. For example, step a) suggests clarifying circumstances, restrictions, and data, but inventive situations starts being difficult precisely because those requirements are unclear. When heuristic is performed by an expert (like Pólya on his book) it seems continuous and easy, but in reality the user generally does not even distinguish the kind of problem he's facing and respond differently when given no evidence and when given worthless evidence, even if the intuitive thinker is an experienced researcher (Tversky & Kahneman 1974).

Heuristics are popular because they are economical and sometimes effective, but they lead to systematic and predictable errors, meaning that in practice they are not reliable for IPS, having similar chances of pointing to the wrong direction than random search.

2.1.2. Brainstorming

The most popular method for generating ideas is Brainstorming (Osborn 1963). Depending on the source (Otto & Wood 2001; Ulrich & Eppinger 2003; Altshuller 1999), the basic rules differ slightly, but without substantial changes. The most important ones are:

- Create a diverse team (different fields) of 6 to 9 people, in a 20 minutes session.
- Focus on quantity of ideas, to increase the chance of producing radical and effective solutions.
- No criticism is permitted, participants should focus on extending or adding to ideas, reserving criticism for a later 'critical stage' of the process.
- Unusual ideas should be analyzed.
- Combine and improve ideas, mixing several to form a single better one.

Two pointers are mentioned in (Dieter 1999):

- Carefully define the problem at the start.
- Allow 5 min for each individual to think through the problem on their own before starting the work.

To diminish the mental inertia, several strategies are proposed, most of them resumed by the acrostic SCAMPER:

- S-Substitute: substitute part of the product/process for something else. How can I substitute the place, time, materials or people?

- C-Combine: What materials, features, processes, people, products or components can I combine?
- A-Adapt: How change the nature of the product/process? What part of the product could I change? What can I copy?
- M-Modify/magnify/minify: Could I add a new twist? Can I change the color, shape, motion?
- P-Put to other purposes: What other market could I use this product in? Who or what else might be able to use it?
- E-Eliminate: It is possible remove a part, function, person without affecting the outcome?
- R-Rearrange/reverse: What if I did it the other way round? What if I reverse the order it is done or the way it is used? How would I achieve the opposite effect? Could I interchange components?

Again, as in heuristics, the first question is how to define the problem. Also even if the problem is well defined, the search space is expanded enormously. The interaction between the team and its ideas can have both beneficial and prejudicial outputs: depending on the circumstances good ideas can be both, developed or diluted by the team.

Brainstorming idea generation process can be resumed as "imagine how you can solve the problem", being its main help being an easy to remember word to the random group idea generation process.

2.1.3. Synectics

Synectics (W. J. J. Gordon 1961) is a problem solving method that takes one step further Brainstorming, integrating diverse individuals into a problem-stating problem-solving group. Its main tool is the analogy, which is classified in 4 types:

- Personal analogy ("empathy"): the user should identify itself with one element in the problem, e.g. a pathologist might identify himself with a virus.
- Direct analogy: The given subject is compared to a similar one of another area. A classical example is Alexander Graham Bell inventing the telephone receiver on the model of the human ear.
- Symbolic analogy: generalization and abstract analogy, it requires an imaginary analogue of the problem. For example, a vision of a snake swallowing its own tail gave the Dutch physicist Kekule a key insight into the benzene molecule.

- Fantasy analogy: the problem-solver projects an image of an ideal solution, reached by magical means or fantastic entities, and then returns to the problem in light of the solution

Success is defined as getting a creative solution that the group is committed to implement.

In Synectics theory a "how" to do the things is presented: analogy. It is still really generic and probably many users will not be able to use it, but it is a potential assistance. However, there is no scientific papers with Synectics as main topic that validate that hypothesis (Science Direct, Google scholar), being mentioned generally in studies of creativity methods during the 70's. It has been compared against *brainstorming* (Bouchard 1972), obtaining a slightly better performance on how psychology students solved 9 different problems (not the typical case of IPS). Regarding the generic activities of IPS, Synectics focuses in creating concepts, almost ignoring the other stages, specially the problem definition, lacking direction from the start.

2.1.4. Conceptual decomposition

The main idea is to decompose a problem in a manner that will be easier to manage, and ultimately provide a greater understanding of it. It is important that every group internal links are deeper than the external links within groups.

There are two main decomposition approaches (Dieter 1999):

- Physical Decomposition: the system is decomposed into subassemblies and components (How it's done).
- Functional Decomposition: the system is decomposed into the required functions to achieve the main one (What needs to be done).

During design the basic functions of the system are considered first, and then each function is further subdivided to provide a better/more detailed understanding. A physical decomposition (parts supporting each function) should be outlined in order to analyze its supporting function.

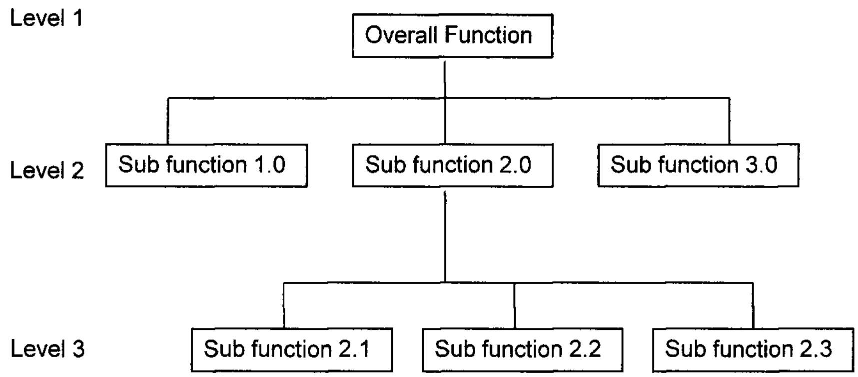

Figure 3: Typical structure of a functional decomposition.

It can be seen that conceptual decomposition is a good previous (or posterior) method to solve IP, since the problem itself is not solved by the decomposition. Some potential negative effects are:

- To focus on the parts, losing the global perspective of what the system needs to achieve.
- Subsystems can "hide" functions and couplings.
- The criteria to decompose the system is essential.
- In user's mind, functions are linked to embodiments (as in the next method analyzed, Morphological analysis.).

2.1.5. Morphological analysis

General Morphological analysis was developed by Fritz Zwicky (1969) as a method for structuring and investigating the total set of relationships contained in multi-dimensional problem, usually non-quantifiable ones. Essentially, five iterative steps define the process (Ritchey 1998):

1. The problem to be solved must be very concisely formulated.
2. All of the parameters that might be of importance for the solution of the given problem must be localized and analyzed.
3. The morphological box or multidimensional matrix, which contains all of the potential solutions of the given problem, is constructed.
4. All the solutions contained in the morphological box are closely scrutinized and evaluated with respect to the purposes that are to be achieved.
5. Fifth step. The optimally suitable solutions are selected and are practically applied, provided the necessary available means.

In its exhaustive search, biases are minimized and new relationships or configurations are not overlooked. It encourages the identification and investigation of boundary conditions, and facilitates communication and group work. Poorly defined parameters also become evident when they are cross-referenced.

It is claimed that "properly applied, morphological analysis offers an excellent balance between freedom and constraints". However that can be said of every method "applied properly".

The difficulties start (again) at the problem definition, which is not assisted neither verified. The parameters definitions are also declared by the user, which is particularly important in inventive situations: how to know the relevant parameters if the *concept* has not been developed yet? In other words, it is necessary a *concept* in order to evaluate parameters, but if we already have a *concept* then the evaluation is artificial, and maybe useless. Morphological analysis seems to be an adequate method for "tasks", but even for them an extensive technical knowledge of the field is mandatory, and psychological inertia can suppress viable alternatives. Finally, the combination of alternatives usually gives an enormous search universe.

2.1.6. Axiomatic design

Axiomatic design (Suh 1998) is introduced as a method to make design more creative, reduce the random solution search process, minimize iterative trial-and-error processes and determine the best design among those proposed. It is presented as a "systematic framework for design because it aids the designer in satisfying multiple design objectives in a homogeneous manner throughout the design process. It is also an effective framework for formalizing and evaluating conceptual design decisions" (Albano & Suh 1992).

Theoretically is based in 2 axioms regarding functional requirements:

- The independence axiom: In an acceptable design, the design parameters (DPs) and the functional requirements (FRs) are related in such a way that a specific DP can be adjusted to satisfy its corresponding FR without affecting other FRs.
- The information axiom: Among alternative designs which satisfy Axiom 1, the best has the minimum information content which means the maximum probability of success[4].

[4] This axiom seems to be inspired in Occam's razor, http://en.wikipedia.org/wiki/Occam%27s_razor .

Both are the basis for several theorems that characterize the bounds on complex systems, considering complexity as a measure of uncertainty in achieving a set of specific functions or functional requirements.

It is stated that whether the design solution is a tangible product, service, software, process, or something else, designers typically follow these steps:

1. Understand their customers' needs
2. Define the problem they must solve to satisfy these needs
3. Create and select a solution
4. Analyze and optimize the proposed solution
5. Check the resulting design against the customers' needs

It can be observed that this list is close to heuristics' one (section 2.1.1), and despite that the two axioms can be potentially helpful in steps 4 and 5, its usefulness cannot be determined for steps 1 to 3, since there is no relation between them (steps 1 to 3 and the axioms). The axioms demand a design, which means (at least) a valid proposal, which is the intended output of steps 1 to 3.

It has been said that axiomatic design is not a design theory as there is no concept and no knowledge described by it, being more appropriated to describe it as command and control theory applicable to some design work (Hatchuel & Weil 2003).

Maybe it is a useful tool to optimize or improve existing systems, like in (Helander & Lin 2000), but regarding the generation of concepts it doesn't provide more assistance than *intuition*.

2.1.7. C-k Theory

The C-K Design Theory was developed by Hatchuel and Weil after a large number of empirical studies (Hatchuel & Weil 2002). It has gained a lot of popularity in recent years, which has helped to its development through contributions and critics.

It aims to provide a unified formal framework for design, including the understanding of the innovative one. Its name reflects the assumption that design can be modeled as the interplay between two spaces with different structure and logic: the space of concepts (C) and the space of knowledge (K). Design then is defined as a process that generates *concepts* from an existing *concept* or transforms a *concept* into *knowledge* (Figure 4). Without the distinction between the expansions of C and K, design disappears or is reduced to mere computation or optimization.

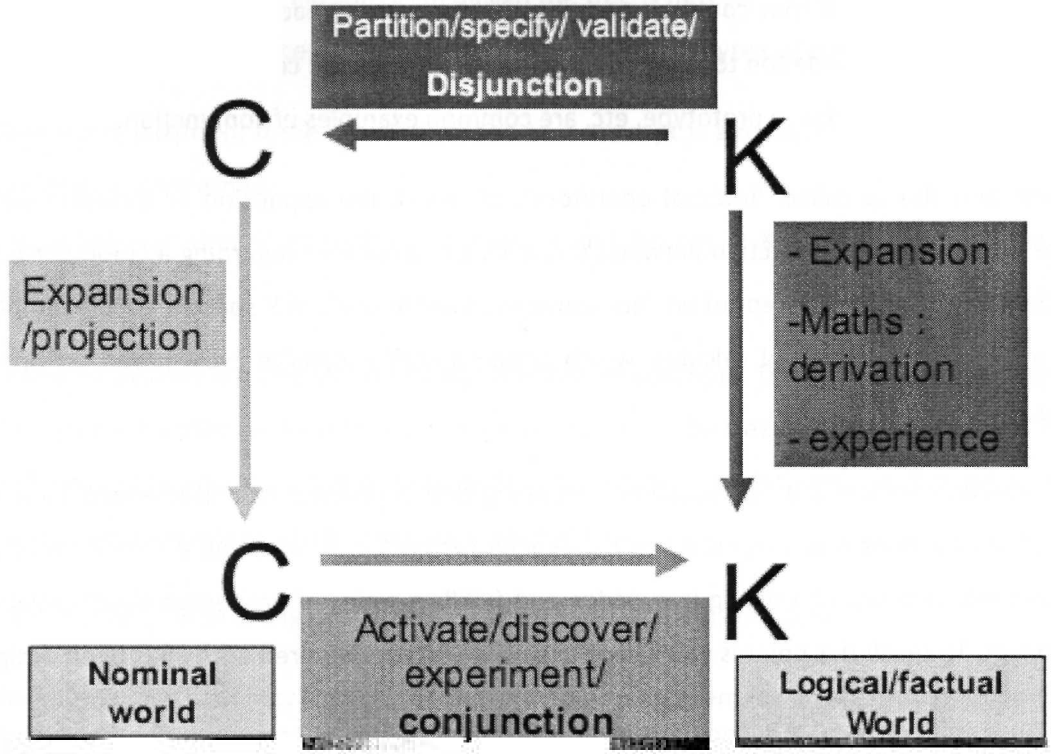

Figure 4: The design square modeled by C-K theory. On it, there are presented the pour types of operators and the fundamental structure of the design process: the independence and interaction between concepts and knowledge.

To define the spaces, C contains ''concepts'' which are undecidable propositions in K, meaning it is not possible to prove that this propositions are true or false in K. Within C, concepts can only be partitioned (into more specific ones) or included (into generic ones). On the other hand K contains all established propositions, meaning all the available knowledge.

The central mechanism of design consists in the process of adding and subtracting properties to concepts or propositions: it can transform propositions of K into concepts of C and conversely. As a group, they are usually called "external operators" and are defined as "disjunction" and "conjunction":

- Disjunction is an operation which transforms propositions going from K to C. This is usually called "generation of alternatives", despite that concepts are not alternatives but potential "seeds" for alternatives. It can be said that the space of C is expanded with elements coming from K.

- Conjunction is an operation which transforms concepts into propositions (from C to K), seeking for properties in K that could be added or subtracted to reach a logical status. In practice it corresponds to validation tools or methods in classical design: consulting an expert, doing a test, an experimental plan, a prototype, etc, are common examples of conjunction.

There exist also the so called "internal operators", on which the expansion or inclusion within each space are the results (no interaction between C and K). E.g. a concept regarding a "modular house" can be partitioned into a new concept of an "economic modular house"; in K space a good example are the rules of logic and propositional calculus, which allows a self- expansive cycle based on proving new theorems.

Usually a design solution is a "conjunction", meaning that a concept is characterized by a sufficient number of propositions that can establish its veracity (or not) in K, reaching a definition of an entity which takes into account all existing knowledge and fulfills a series of properties clearly related to the initial concept. To reach this point is equivalent to saying that the required entity has been designed.

Another illustration of the C-K dynamics is given in Figure 5, where is presented the tree structure in C, situation not necessary in K. It is also presented that any expansion in C is dependant of K and reciprocally. Any choice to expand or not in C is K-dependant. Conversely, any creation in K requires travelling by some path in C. Designs begins with a disjunction, and finishes only if some conjunction exists and is judged as a solution relative to K.

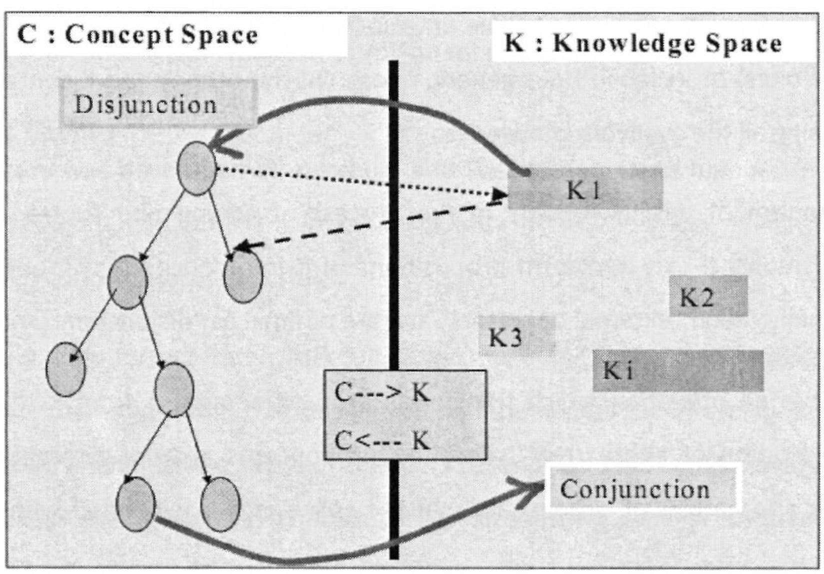

Figure 5: C-K dynamics

A point worth of being clarified is that adding properties to a concept seems to decompose a concept into sub-concepts, being this an illusion, as in design the tree is necessarily an expansion of the concept. To understand this point it is necessary to distinguish between two types of partitions:

- Restricting partition: If the property added to a concept is already known in K as a property of the entities concerned.
- Expanding partition: if the added property is not known in K as a property of the entities concerned.

Creativity and innovation are due to expanding partitions of concepts: This also reveals why creativity is built in their (the C-K authors) definition of design: concepts can be freely expanded provided there are available expanding properties. But those properties can only from K, showing how the unknown comes from what is already known and that concepts are the vehicle.

C-K theory has several interesting points to this dissertation, standing out two:

- Any design methodology that can be performed as a program (or an algorithm) without any use of concepts and C-sets is finally reduced to a K-K operator. This includes GAs or any other algorithms for optimizing engineering systems that uses only standard calculus and logic (Hatchuel & Weil 2009).
- It is avoided the classical logic of design stages from "abstract to concrete" or from "rough to detail". These are too normative positions: "details" may come first in a design if they have a strong partitioning power ;.and unexpected stages could result from a surprising knowledge expansion. The classical opposition between linearity and turbulence disappears: innovations could result from both.

However, its proposal is focused on the description of design, being its main utility the clarification of the problematic situation. Its capacity to generate valid proposals in inventive situations has not been shown so far.

2.1.8. Concept Generator

Presented in the article "A computational approach to conceptual design" (Strawbridge et al. 2002), the *Concept Generator* claims that it can assist a designer at the conceptual stage of design. Developed from an empirical study of consumer products, its goal is to produce many concepts quickly and early in the design process by making use of existing design knowledge, reducing the number of iterations in the design process. A new (or redesigned) product's functional model is the input into the *concept generator* and product components are the output (Figure 6).

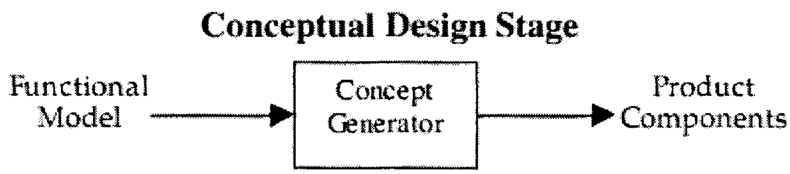

Figure 6: Concept generator flow chart

The article recognizes that "the concept generation, has seen less generally applicable tools even though well known methods are available (e.g., brainstorming, intrinsic and extrinsic searches, morphological matrices, etc.)", being that the main reason why the method was selected to be analyzed.

On its conclusions, the article claims that concept generator helps to quickly obtain product concepts, the design process is made easier, quicker and does not require the efforts of an entire team, demonstrating that the creation of a morphological matrix is a computable process. It can be seen that having an optimistic point of view, the only improved characteristic is the "speed", showing that the method is really a computational implementation of *Morphological analysis* (section 2.1.5), and thus has all of its problems. In general, its *ideality (*sum of benefits that provides divided by the sum of all costs and harms) is low, because gives no means to define neither evaluate the original problematic.

2.1.9. TRIZ

TRIZ is a Romanized acronym for the Russian "Теория решения изобретательских задач", meaning "Theory of Inventive Problem Solving". It has emerged as a problem-solving theory and methodology based on the study of solution patterns of real-world problems (since 1946), under the hypothesis that creative innovations are based upon universal principles that can be identified, codified, and then be taught to make the process of creativity more predictable. Behind the previous statements lies the belief of TRIZ founder, G.S. Altshuller, that creativity could be turn into an "exact science", as it can be appreciated in the following words: "Although people who have achieved a great deal in science and technology talked of the inscrutability of creativity, I was not convinced and disbelieved them immediately without argument. Why should everything but creativity be open to scrutiny? What kind of process can this be which unlike all others is not subject to control?" (Altshuller 1984).

The research has proceeded in several stages during the last sixty years. The three primary findings of the studies are (Barry et al. 2008):

- Problems and solutions are repeated across industries and sciences. The classification of **the contradictions in each problem predicts the creative solutions** to that problem.
- Patterns of technical evolution are repeated across industries and sciences.
- Creative innovations use scientific effects outside the field where they were developed.

Nowadays, TRIZ theory has proved its utility through many practical applications, publications, and a sustained growing, being one of its most successful tools the "40 Inventive Principles" (40-IP).

They consist in a group of generic solutions that solved technical contradictions across many fields, which were deducted ("distillated") after analyzing thousands of patents (Malmqvist et al. 1996). The process of solving problems using the 40-IP consists basically in:

- Convert your specific problem into a general one
- Look at the general solutions proposed
- Translate it into a specific solution to your problem

A diagram about this process is presented in Figure 7.

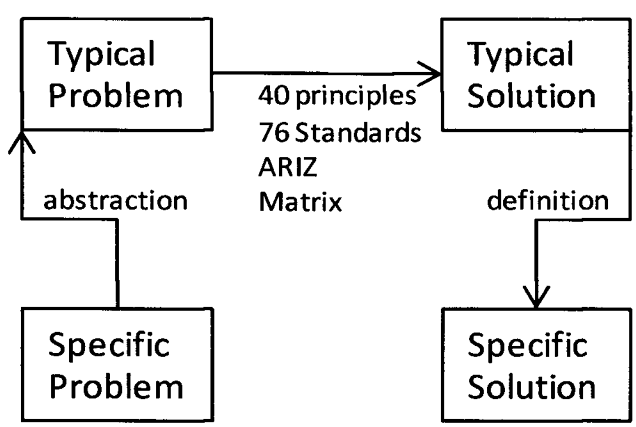

Figure 7: How TRIZ works.

The 40-IP (seeTable 1) are, in Altshuler's words, "40 basic methods of eliminating contradictions". They became popular mainly because of the "contradiction matrix" (Altshuller & Shulyak 1998), on which the principles and the contradiction that they solve are placed in a format easy to handle, even for beginners.

The principles and the matrix, together with the other tools showed in Figure 7, are usually referred as part of "Classical TRIZ", in order to distinguish them from the work that many researchers have done individually, starting from TRIZ ideas.

Both the 40-IP and the contradiction matrix have been subjected to several attempts of improvement. For example, in 2003 it was presented the "2003 Matrix": created by several TRIZ experts (Mann et al. 2003) it consists in an enhanced version of the classical contradiction Matrix that It also extends the inventive problem to 76, including the most popular combinations of the 40-IP as new ones.

Table 1: The "40 inventive principles". Definitions and examples in (Altshuller 1984; Malmqvist et al. 1996)

1. Segmentation	11. Beforehand cushioning	21. Skipping	31. Porous materials
2. Taking out	12. Equipotentiality	22. "Blessing in disguise"	32. Color changes
3. Local quality	13. 'The other way round'	23. Feedback	33. Homogeneity
4. Asymmetry	14. Spheroidality - Curvature	24. 'Intermediary'	34. Discarding and recovering
5. Merging	15. Dynamics	25. Self-service	35. Parameter changes
6. Universality	16. Partial or excessive actions	26. Copying	36. Phase transitions
7. "Nested doll"	17. Another dimension	27. Cheap short-living objects	37. Thermal expansion
8. Anti-weight	18. Mechanical vibration	28. Mechanics substitution	38. Strong oxidants
9. Preliminary anti-action	19. Periodic action	29.Pneumatics and hydraulics	39. Inert atmosphere
10. Preliminary action	20. Continuity of useful action	30. Flexible shells and thin films	40. Composite materials

Regarding its effectiveness, a study compared the prediction rate of the 2003 Matrix versus the classical one (Mann 2004). The experiment consisted in select a group of innovative patents and analyze if the utilized solution was able to be predicted by using the matrixes. The prediction rate of the classical Matrix was around a 30%, versus a 96% of the 2003 Matrix. The study (as the author recognizes) can be criticized in several aspects, like potential biases, people who analyze the patents, assumptions made, the selective criteria to classify a patent as innovative, etc, but so far no other study about the topic has been conducted.

Another example was found in the industry: at Air Products. When they get into TRIZ it was "agreed that some definitions are somewhat ambiguous and hard to understand. It was also noticed that some principles are very specific, often related to specific mechanical devices and industrial applications which simply aren't relevant to our organization". What they did was to group the 40-IP into five categories (Brostow 2006):

1. Compensate or prepare for.
2. Change appearance, structure, composition, condition.
3. Do the opposite, different, take to extreme.
4. Make the same, use together, combine.
5. Separate from, remove, disturb.

The validity of their approach is out of scope, but it is important to notice the personalization of the methodology and the tendency to group similar concepts into more versatile branches. They claim that about 5 to 10 principles allow reaching most of the solutions, which is at least debatable, but if the company solves its problems through this approach, is worth to be analyzed.

That is similar to what Ford Motor Company did in 1995, adapting SIT (presented in section 2.1.10) developing a methodology named USIT (Unified Structured Inventive Thinking). The goals and characteristic features of USIT may be summarized as follows (Nakagawa 2000):

- It concentrates in the "Concept Generation Stage", which requires the most of creativity, and hence has been supported very poorly by conventional methods.
- USIT intends to be applied to real practical problems for rapidly generating multiple conceptual solutions. It goal is to solve real problems in industries, and does not put emphasis on amazing inventions.

- It provides a defined procedure for applying the methodology, composed of three stages, i.e. problem definition, problem analysis, and solution generation, as illustrated in Figure 8.
- The technological system of the problem is described in clearly-defined terms of Objects, Attributes, and Functions.
- Elements of techniques in USIT are simple and explained in guidelines. Especially, there are only four techniques used for the solution generation, and they are used in correspondence to the concepts of objects, attributes, and functions.
- No outside knowledge bases and software tools are used in USIT. It should be learned and memorized.
- Engineering details, such as specifications, figures, numbers, costs, deadlines, etc, are put aside the consideration during the USIT procedure. The multiple conceptual solutions obtained with USIT should later be reviewed from various engineering and business points of view introducing the above mentioned engineering details.

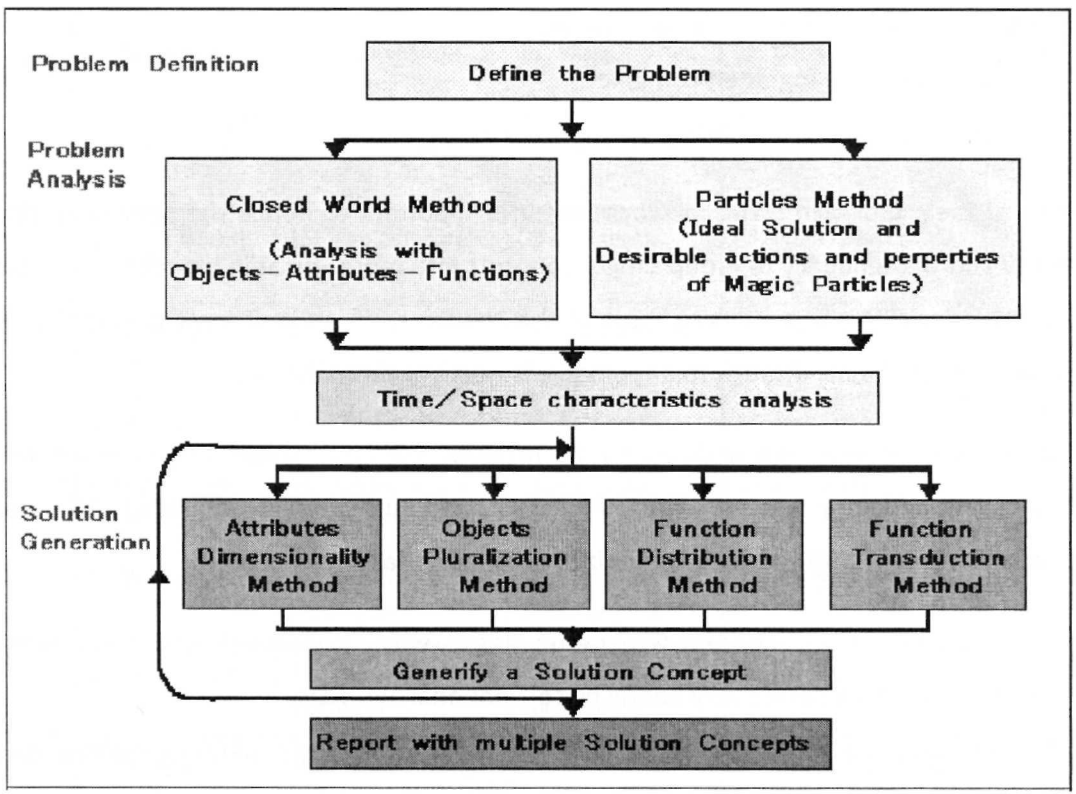

Figure 8: Flowchart of the USIT procedure.

Following this structure, Nakagawa has proposed the 'Six-Box Scheme of Creative Problem Solving' methodology (Nakagawa 2006), which has several implications in how to perform the whole process of IPS (Figure 9). As it can be seen, there is an ongoing debate about the relative benefits and evolution of the different TRIZ tools.

In general, several observations and critics can be made to TRIZ tools in general and to the 40-IP in particular: they have a non-logical sequencing, are overlapped, with different abstraction level, and are difficult to remember (Mann 2002). Additionally, their presentation format does not allow deducting information, to delete repeated or obsolete principles, figure out "patterns" or "zones" of solution, etc.

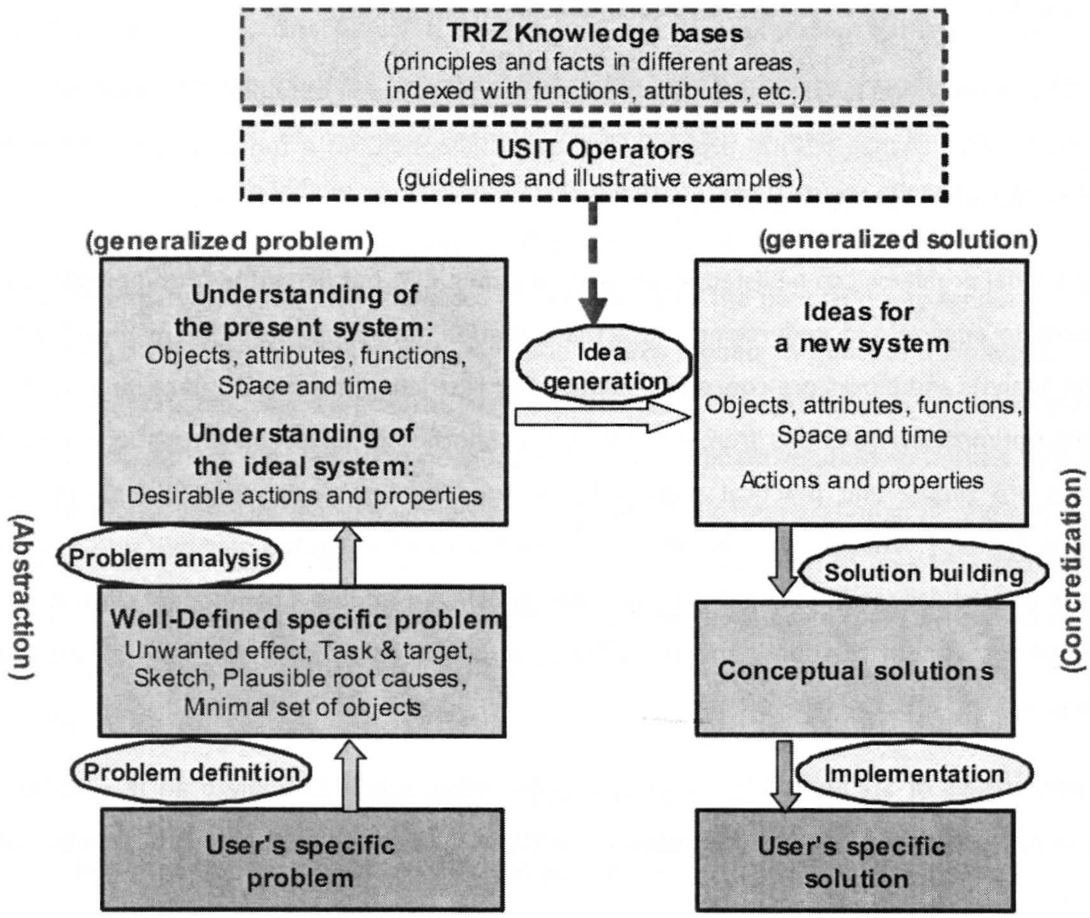

Figure 9: USIT six-box scheme.

Despite the previous, TRIZ offers a concrete proposal to solve problems, and the 40-IP have proven to be more than random ideas, which could be used to effectively aid problem solving.

2.1.10. SIT

Structured Inventive Thinking (SIT) origin can be found in TRIZ theory, which was gradually modified through experience, theory, and experimental study, until the method changed to become a different one (Horowitz & Maimon 1997). It was not included in the previous section because at the present is a complete different theory, and also its particular characteristics are interesting for this research.

Their deepest differences are first, that the notion of overcoming conflicts is replaced with the application of the sufficient conditions, and second that SIT applies a minimal set of techniques that should be used implicitly after some training.

To determine if a solution is creative, two conditions must be fulfilled:

- The closed world (CW) condition: no new element is added in obtaining the solution.
- Qualitative change (QC) in problem characteristic condition: a fundamental relation between variables has to change qualitatively.

Many potential problems can be listed, especially regarding CW, but like none of the previous methods it demands to analyze and understand the actual situation, and thus the problem: the CW condition force the analysis and blocks pre-conceived solutions by restricting the search space, and the QC clearly separates optimization solutions from creative ones. Another positive characteristic is that the two conditions are simple and practical enough to be used by any design practitioner, in an almost challenging format ("What can you do with these 2 restrictions?"). On the negative side, the definition of CW can discard easy to implement solution (like changing or adding a component), it is necessary to understand the system relations, and the criteria to select parameters can lead to pre-conceived solution spaces.

The characteristics of SIT approach makes it an interesting option to analyze an IP situation before utilizing other methods or compromise resources without understanding the most basic elements to be considered.

2.2. Evolutionary algorithms and the creative process

The literature linking EAs and the IPS process is not as clearly defined-grouped as in the creative design methodologies. However, between them, Genetic Algorithms (GAs), Genetic Programming (GP), and Evolutionary Design (ED) stand out.

2.2.1. Genetic algorithms

GAs are part of evolutionary computing. They are basically blind trial and error procedures that assign values to function parameters, inspired in the evolution process observed in nature. In sexual reproduction, each parent's genes—combined with random genetic mutation—create organisms with new characteristics, and the fittest ones have higher possibilities of surviving and passing on their genes to succeeding generations. GAs follow that idea within a computational framework, acting on a population P(t) of candidate solutions for exploration in the search space by introducing variations into the population by means of idealized genetic recombination operators. The most important recombination operator is called *crossover* (Table 2). By means of the crossover operator, two structures in the new population exchange portions of their internal representation. Additionally, there is *mutation*, a secondary operator that increases the variability of the population by randomly changing each bit position of the structure in the new population with a probability equal to the mutation rate M. During each iteration step, called a *generation*, the structures in the current population are evaluated, and, on the basis of those evaluations, a new population of candidate solutions is formed by means of the recombination operators using the individuals of the former generation that showed the best performance. Then the edited "survivors" constitute the new generation to be processed. This cycle continues until a certain criterion is reached. Experimental studies indicate that GAs exhibit extremely high efficiency, consistently outperforming both gradient techniques and various forms of random search (Coello 2000).

Table 2. An example of crossover

Parent Chromosomes (potential solutions)	Children, if cross point is the third position	Children, if cross point is the sixth position
1 1 1 1 1 0 0 0 0 0	*1 1 1* 0 0 1 1 1 1 1	*1 1 1 1 1 0* 1 1 1 1
0 0 0 0 0 1 1 1 1 1	0 0 0 *1 1 0 0 0 0 0*	0 0 0 0 0 1 *0 0 0 0*

The generic GA cycle can be summarized as follows:

1. **[Start]** Generate random population of n chromosomes (suitable solutions for the problem)
2. [**Fitness**] Evaluate the fitness f(x) of each chromosome x in the population.
3. [**New population**] Create a new population by repeating following steps:
    a. [**Selection**] Select two parent chromosomes from a population according to their fitness (the better the fitness, the greater the chance to be selected).
    b. [**Crossover**] With a crossover probability, crossover the parents to form new offspring (children). If no crossover was performed, offspring are an exact copy of the parents.
    c. [**Mutation**] With a mutation probability, mutate new offspring at each position in the chromosome.
    d. [**Accepting**] Place new offspring in a new population.
4. [**Replace**] Use the new generated population for another run of the algorithm.
5. [**Test**] If the end condition is satisfied, stop and return to the best solution so far.
6. [**Loop**] Return to b)

Over the generations it is expected that the results will converge towards an optimum (Figure 10). The size of the population and number of generations are critical to achieve this task, however, to increase them implies more computational time.

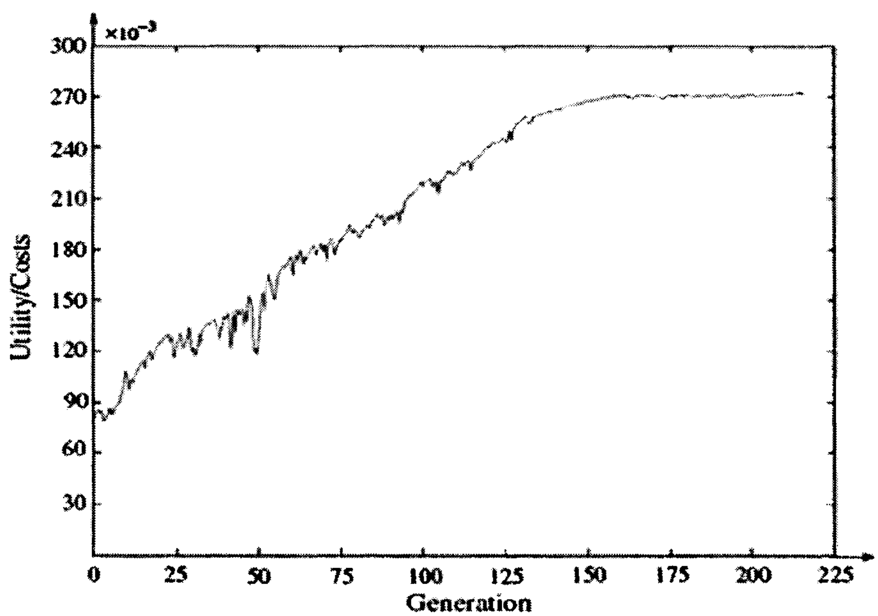

Figure 10.An example of evaluation improvement throughout generations **(Jiao et al. 2007)**

In order to be operational the GA needs a finite representation of the solutions (not the solutions themselves), that it must be decided at the beginning of the process, together with the means or procedure for discriminating good solutions from bad solutions. This can be as simple as having a human choosing the best solutions over the worse ones, or it can be an elaborate computer simulation or model that helps determine what good is, but something must determine a solution's relative fitness to purpose.

Although GA have proven to be efficient in converging to optimal values in the given search space, they are not free from being stopped at Pareto fronts that avoid achieving "ideal" results.

2.2.2. Genetic programming

Genetic programming is a systematic method for getting computers to automatically solve problems. It starts from a high-level statement of what needs to be done and automatically creates a computer program to solve the problem by means of a simulated evolutionary process (J. R. Koza et al. 2008). In his articles, Koza claims that GP:

- Routinely delivers high-return human-competitive machine intelligence
- Is an automated invention machine
- Can automatically create a general solution to a problem in the form of a parameterized topology
- Has delivered a progression of qualitatively more substantial results in synchrony with the augmentation of computational power.

In order to solve a problematic, the user must communicate the characteristics of the situation to the GP system. The five major preparatory steps for GP entail determining:

- The set of terminals available to each branch of the to-be-evolved computer program (e.g., the independent variables of the problem, no-argument functions, and random constants)
- The set of primitive functions available to each branch of the to-be-evolved computer program,
- The fitness measure (for explicitly or implicitly measuring the fitness of individuals in the population),
- Several parameters for controlling the run
- The termination criterion and method for designating the result of the run.

The process is similar to GA:

1. **[Start]** An initial population (random) of functions and terminals is generated.
2. **[Fitness]** Every member of the population is evaluated using the fitness measure
3. **[New population]** Create a new population by repeating following steps: on the selected programs, with a probability based on its aptitude:
    a. **[Selection]** Select two parent chromosomes from the selected programs, according to their aptitude (the better the aptitude, the greater the chance to be selected).
    b. **[Crossover]** Create a new offspring program for the new population by recombining randomly chosen parts of two selected programs.
    c. **[Mutation]** Create one new offspring program for the new population by randomly mutating a randomly chosen part of the selected program.
    d. **[Architecture-altering operations]**: Create one new offspring program for the new population by applying a selected architecture-altering operation to the selected program.
    e. **[Accepting]** Place new offspring in a new population.
4. **[Replace]** Use the new generated population for another run of the algorithm.
5. **[Test]** If the end condition is satisfied, stop and return to the best solution so far.
6. **[Loop]** Return to b)

GP can be applied to problems in a variety of fields, including design ones. By 2004, there were at least 22 known instances where GP has duplicated the functionality of a previously patented invention or infringed a previously issued patent, as showed in Table 3.

Table 3: Twenty-two previously patented inventions reinvented by GP (J. Koza et al. 2004).

Invention	Date	Inventor	Place	Patent
Mechanical system composed of rigid members for drawing a straight line without reference to another straight line	1841	Robert Willis	Great Britain	British 6258
Ladder filter	1917	George Campbell	AT&T	US 1,227,113
Crossover filter	1925	Otto Julius Zobel	AT&T	US 1,538,964
"*M*-derived half section" filter	1925	Otto Julius Zobel	AT&T	US 1,538,964
Cauer (elliptic) topology for filters	1934–1936	Wilhelm Cauer	University of Gottingen	US 1,958,742, US 1,989,545
Negative feedback	1937	Harold S. Black	AT&T	US 2,102,670, US 2,102,671
Proportional, integrative, and derivative controller	1939	Albert Callender & Allan Stevenson	Imperial Chemical Limited	US 2,175,985
Second-derivative controller	1942	Harry Jones	Brown Instrument Co.	US 2,282,726
Darlington emitter–follower section	1953	Sidney Darlington	Bell Telephone Laboratories	US 2,663,806
Philbrick circuit	1956	George Philbrick	George A. Philbrick Research	US 2,730,679
Sorting network	1962	Daniel G. O'Connor & Raymond J. Nelson	General Precision, Inc.	US 3,029,413
NAND circuit	1971	David H. Chung & Bill H. Terrell	Texas Instruments Inc.	US 3,560,760
Computational circuits	Numerous	Numerous	Numerous	Numerous
Electronic thermometer	Numerous	Numerous	Numerous	Numerous
Voltage reference circuit	Numerous	Numerous	Numerous	Numerous
60- and 96-dB amplifiers	Numerous	Numerous	Numerous	Numerous
Cubic function generator	2000	Stefano Cipriani & Anthony A. Takeshian	Conexant Systems, Inc.	US 6,160,427
Mixed analog–digital variable capacitor circuit	2000	Turgut Sefket Aytur	Lucent Technologies Inc.	US 013,958
Voltage–current conversion circuit	2000	Akira Ikeuchi & Naoshi Tokuda	Mitsumi Electric Co., Ltd.	US 6,166,529
Low-voltage balun circuit	2001	Sang Gug Lee	Information and Communications University	US 6,265,908
High-current load circuit	2001	Timothy Daun-Lindberg & Michael Miller	IBM	US 6,211,726
Tunable integrated active filter	2001	Robert Irvine & Bernd Kolb	Infineon Technologies AG	US 6,225,859

It is important to clarify that a fitness measure is necessary, but this measure is not necessarily a measure of the "innovation" of the individual, but the degree of proximity towards a pre-defined objective. Thus, the objective is the one that needs a strong definition.

GP is a promising approach for achieving automated design because several non biased steps are inherent in the evolutionary process. It can be criticized because of the limitations defined by the terminals and functions, meaning that the search space is restricted a priori. However, adding a few unnecessary primitives in an attempt to ensure sufficiency does not tend to slow down GP overmuch, although there are cases where it can affect the system in unexpected ways (Poli et al. 2008). This characteristic can be considered and advantage whenever is necessary to work modifying (generally expanding) the search space.

2.2.3. Evolutionary design

Considered a sub-field of evolutionary computation, *evolutionary design* is directly related to engineering design problems. It has been described like a process capable of generating designs by changing topologies and shapes, allowing this way that an intricate design can arise through a slow, gradual improvement process (Bentley 1999).

Like the EAs, it integrates ideas from computer science and evolutionary biology, but also from engineering design science. Four of the major categories of problems considered by evolutionary design include evolutionary design optimization, creative evolutionary design, evolutionary art, and evolutionary artificial life forms (Kicinger et al. 2005).

For the analysis it is considered that engineering design process performed by a human designer can be generally summarized as consisting of the following stages:

1. Preliminary or conceptual design
2. Detailed design
3. Evaluation
4. Iterative redesign, if the evaluation results are unsatisfactory

Bentley makes the observation (still valid) that computers have been used successfully for all of these stages except the first: conceptual or creative design. As stated by Goldberg, "the creative processes of engineering design have long been regarded as a black art. While the engine of analysis steamrolls ever forward, our understanding of conceptual design seems locked in a timewarp of platitudes, vague design procedures, and problem-specific design rules" (D. E Goldberg 1991).

To research the possibilities of evolutionary design, Bentley applied the system to a task considered 'hard' for a genetic algorithm: the design of optical prism, which consist of the generation of appropriate geometries for optical prisms to allow light to be directed through them according to various design specifications (Bentley & Wakefield 1997a). The system is generic, i.e. capable of the creation and optimization of the geometry of a wide range of three dimensional solid objects, and is based on three elements:

1. A suitable representation of solid objects to allow the computer to manipulate candidate designs effectively during the design process.
2. A modified genetic algorithm to evolve such represented designs from scratch.

3. Evaluation software to guide the evolution process

The representation is a critical issue, since it requires fully describe the complete shape, and to be able to be modified in any way by the genetic algorithms. This situation differs considerably from the requirements where representations are to be manipulated by a human.

The final representation utilized combined ideas from constructive solid geometry (CSG) and standard spatial partitioning methods, by allowing every partition, or primitive, to vary in size and position, and to be intersected by a plane of variable orientation (Figure 11). When used in combination, those primitives are capable of closely approximating any 3D (and 2D) solid object whilst requiring very few definition parameters.

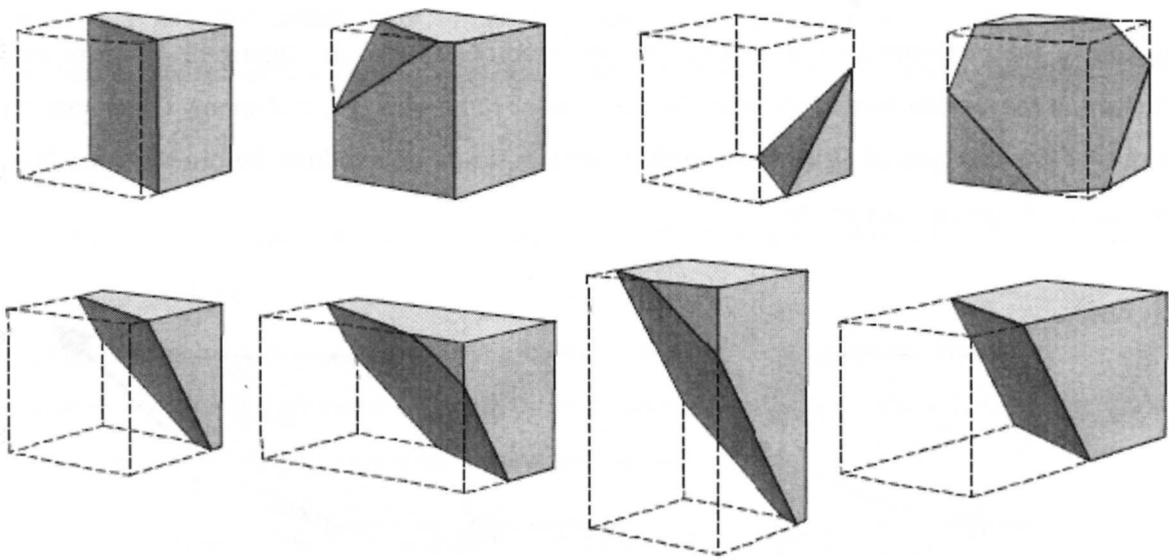

Figure 11: Clipped stretched cubes.

The direction of light through an optical prism, is considered de*ceptive* for GAs, because there exists many accessible local minima (*deceptive attractors)*, and a single, hard to reach global minima.

To illustrate the previous, it can be considered a rhomboid prism acting like periscope through a double reflection (Figure 12). If either the top or bottom reflection is absent (Figure 12, b and c), or if the two primitives making up the design are not correctly aligned (Figure 12 d), the design fails. In other words, every part of the design depends on the others completely.

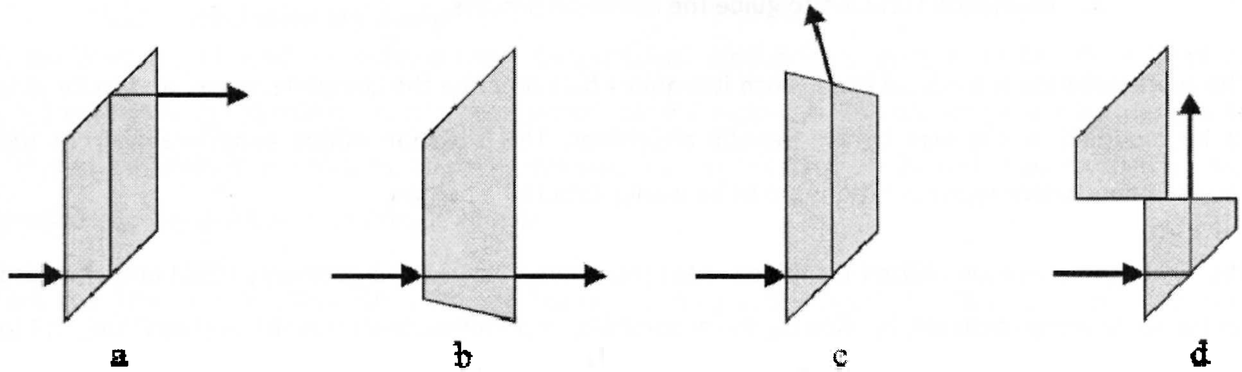

Figure 12: Correct rhomboid prism (a), failed rhomboid prism attempts (b,c,d).

Although designs such as those shown in Figure 12 b and c could be combined to form a perfect rhomboid prism, the GA would rarely pick those designs for reproduction, since they are very unfit compared to the potential deceptive attractors to the problem, like in Figure 13. Because of their simplicity and reasonable fitness, the attractors are easier to be selected, and so any GA almost always ensures that it is this type of design that predominates in future populations, before the good features of the less fit designs can be exploited.

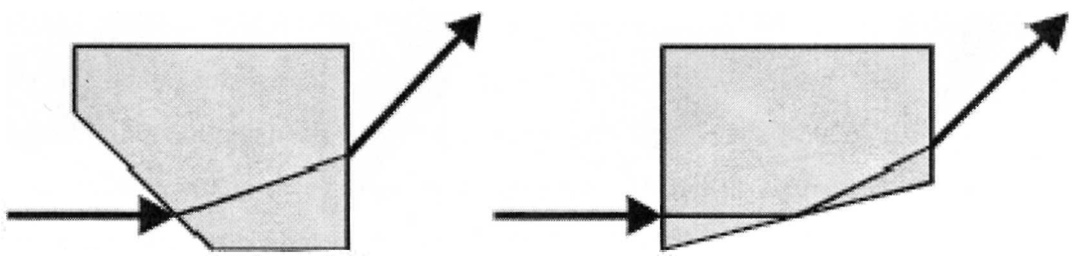

Figure 13: Two deceptive attractors to the 'rhomboid prism' problem.

The results of the study were good since the analyzed problems were solved despite its deceptive nature. To evolve new conceptual designs from scratch the system created numerous types of prism successfully, either performing the whole design process entirely itself, or assembling new designs out of smaller, previously evolved components, with the advantage that a computer is not limited by 'conventional wisdom', so it could generate original designs based on entirely new principles.

For simple types of prism (i.e., prisms with at most two total internal reflections) most problems could be overcome by giving more information, like additional requirements (symmetry, no refraction, etc). This constrained the range of acceptable solutions, reducing the deceptive attractors and increasing the probability of evolving good designs. More complex types of prism (i.e., prisms with four total internal reflections) needed more that adding information to evolve designs, so the problem was simplified using previously created components.

At this stage it can be appreciated a conflict: increasing the specification in primitives diminishes the "deception", but it blocks the search space and increase the complexity of the combinations. A potential strategy is first to evolve the components, and then its placement.

Like in the previous example, the use of GAs combined with CAD and analysis software has proven to be useful for searching the design space for better solutions (Bentley 1999; Albers et al. 2009), and to explore also the nature of EAs, although the exact approach used by developers of such systems varies.

2.3. More specific computational concept development studies, related to CAD or EAs

During the last years several ideas regarding conceptual design have been developed. In this section, eight articles are briefly presented to put in context what is the status of these hybrid approaches to solve problems based in evolutionary computing and/or CAD systems.

2.3.1. Impact of CAD tools on creative problem solving

During the presentation of the results of a survey applied to a group of CAD users, Robertson analyzes how computers may influence the ability to design creatively (Robertson & Radcliffe 2009). Four mechanisms by which CAD tools may influence the creative problem solving process were extracted from a series of observations (``critical incidents") during the case study: enhanced visualization and communication, circumscribed thinking, bounded ideation, and premature fixation.

1. **Enhanced visualization and communication**: the use of CAD in the project greatly enhanced the ability of the team to visualize and communicate their ideas. However, some subtleties emerged from the data, particularly it is interesting to compare the relative use of different modes of working between mature and immature designs, and between visualization and communication.

For simple types of prism (i.e., prisms with at most two total internal reflections) most problems could be overcome by giving more information, like additional requirements (symmetry, no refraction, etc). This constrained the range of acceptable solutions, reducing the deceptive attractors and increasing the probability of evolving good designs. More complex types of prism (i.e., prisms with four total internal reflections) needed more that adding information to evolve designs, so the problem was simplified using previously created components.

At this stage it can be appreciated a conflict: increasing the specification in primitives diminishes the "deception", but it blocks the search space and increase the complexity of the combinations. A potential strategy is first to evolve the components, and then its placement.

Like in the previous example, the use of GAs combined with CAD and analysis software has proven to be useful for searching the design space for better solutions (Bentley 1999; Albers et al. 2009), and to explore also the nature of EAs, although the exact approach used by developers of such systems varies.

2.3. More specific computational concept development studies, related to CAD or EAs

During the last years several ideas regarding conceptual design have been developed. In this section, eight articles are briefly presented to put in context what is the status of these hybrid approaches to solve problems based in evolutionary computing and/or CAD systems.

2.3.1. Impact of CAD tools on creative problem solving

During the presentation of the results of a survey applied to a group of CAD users, Robertson analyzes how computers may influence the ability to design creatively (Robertson & Radcliffe 2009). Four mechanisms by which CAD tools may influence the creative problem solving process were extracted from a series of observations (``critical incidents") during the case study: enhanced visualization and communication, circumscribed thinking, bounded ideation, and premature fixation.

1. **Enhanced visualization and communication**: the use of CAD in the project greatly enhanced the ability of the team to visualize and communicate their ideas. However, some subtleties emerged from the data, particularly it is interesting to compare the relative use of different modes of working between mature and immature designs, and between visualization and communication.

For immature designs, there was a lower level of CAD usage and more use of three other modes of working: free hand sketching, verbal discussions, and drawings boards. **The inference is that CAD is a tool that is better suited for detailed design than for conceptual design.**

2. **Circumscribed thinking**: CAD tools often limited the solutions available to the team, not only to what is possible with a determined package (its technical limitations), but what is easiest, especially when there exists time pressures. This tendency to privilege the easiest over the best is a potential barrier to creative design, until the moment that the designer augments its proficiency and exploits the power and functionality of the CAD aiming for ``perfection". Both extremes are referred as circumscribed thinking, and lead to the waste of resources, being positive or negative.
3. **Bounded ideation**: It was observed on the studied project that more ideas were generated by the team members who did not use the advanced 3D CAD tools. Furthermore, the best environment for idea generation tended to occur away from computers, in small meetings, characterized by large amounts of sketching and discussion. The intrinsic motivation of the designer has a central role to play in promoting creativity
4. **Premature design fixation:** the more detailed CAD models (closer to be finished) developed the tendency to reject ideas which would lead to too many changes to the model itself or to its underlying structure. This design fixation existed even if the proposed changes solved numerous problems or make other improvements (like reducing the overall risk). This design fixation was strongly dependent of the schedule.

The survey itself aims to discover whether the phenomena identified in the case study were experienced more generally by engineering designers who use CAD. The case study involved the collection of in-depth, qualitative data, and the survey provided the opportunity to test those findings in other situations.

The survey was targeted specifically at engineering designers who regularly use mechanical 3D CAD packages in their work. The most relevant results are presented in the next images:

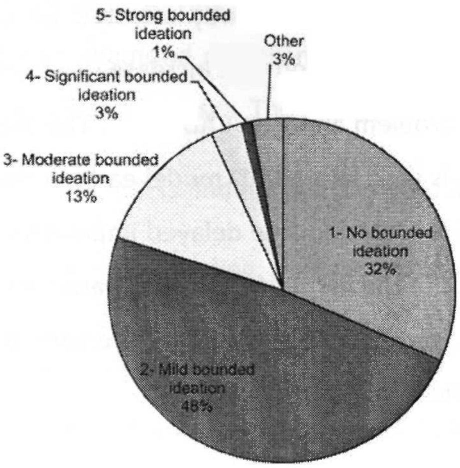

Figure 14: extent of bounded ideation.

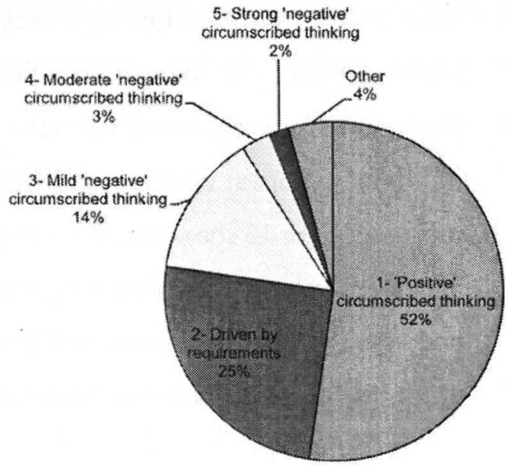

Figure 15: Circumscribed thinking question.

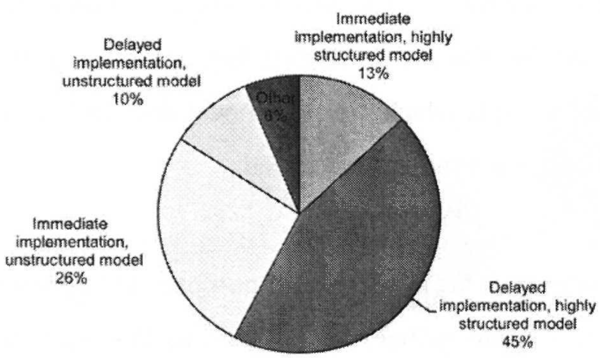

Figure 16: Precursors of premature fixation.

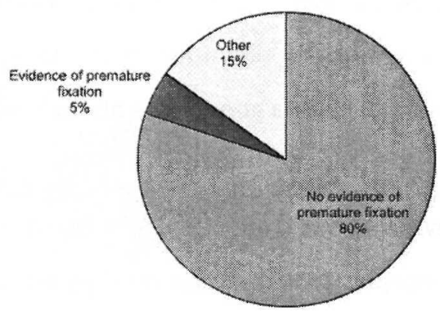

Figure 17: Extent of premature fixation.

Several findings were made:

- Regarding communication and visualization, it was discovered that a good practice is to avoid modeling in the concept stage: it is often used in an unstructured way, and encourages individual work over group work. Two groups of users (inexperienced and those who use CAD constantly) seem to be particularly susceptible, however, no strong conclusions are presented.
- In the ideal situation circumscribed thinking does not arises: a designer is constrained only by the requirements of the task and is free to express their intent. As shown in Figure 15, roughly half of the designers present positive circumscribed thinking, going beyond merely satisfying the requirements and 19% admitted that the design could be negatively influenced.

- The survey data indicates that most CAD users are not significantly affected by bounded ideation (Figure 14). There are however periods of low motivation that increase its effect.
- Premature fixation is a real, but not a widespread problem amongst CAD users. The precursors of it tend to appear when a high level of structure is built into a CAD model early in the design process. Figure 16 show that most respondents (45%) opted for the delayed implementation of a highly structured model implying that the features of the packages involving parameterization and interconnections are avoided in the earlier stages of design. Figure 17 presents that only 5% of the respondents chose the option which implied premature fixation.

The results provide a basis for advising the developers of CAD tools on ways to enhance systems for use during the conceptual phase of product development, together with advice for designers in how to foster creativity avoiding possible pitfalls in the use of CAD tools. Those virtues are also related to two problem founded on the article: it consistently sees CAD as a support for designer, and its survey nature implies that the own opinion of the users is too relevant. Despite the previous objections and that they utilize an ambiguous definition of creativity (ideas or concepts which are both novel and useful), the output is considered a good guide about how CAD is effectively used and visualized.

2.3.2. Creativity for problem solvers

In his review about creativity, Vidal (Vidal 2009) declares that his papers' main purpose is "to present some concepts, tools and approaches from the broad interdisciplinary field known as Creativity and Participative Problem Solving that will enrich the toolbox of problem solvers". To do the previous, some modern and interdisciplinary concepts about creativity and creative processes are presented, fundamental publications related to the theme are briefly reviewed, and a generic approach is explained.

The definition of creativity is the starting point, consisting in (among other things) "the ability to challenge assumptions, recognize patterns, see in new ways, make connections, take risks, and seize upon chance". Challenge assumptions to question the basis of the problem formulation; recognise patterns because they lead to the solution; see in new ways to find patterns from different perspectives: a rational or logical, an organizational or procedural, an interpersonal or emotional, and an experimental or holistic; make connections to build from two thoughts or perceptions; take risks because there always exists the probability that ideas will lead to failure; and seize upon a chance to take advantage of an opening that allows to move forward toward a creative solution.

Creative problem solving (CPS) is presented utilizing the 6-diamond model (Figure 18) where the upper part of each diamond represents the divergent sub-processes and the lower part corresponds to the convergent sub-processes. The six steps consist in:

- Mess finding: fluency, flexibility, originality, deferred judgment and evaluation.
- Fact finding: analysis and evaluation.
- Problem finding: synthesis.
- Idea finding: fluency, flexibility, analysis, originality and deferred judgment.
- Solution finding: synthesis, elaboration and evaluation.
- Acceptance finding: synthesis, evaluation, originality and flexibility.

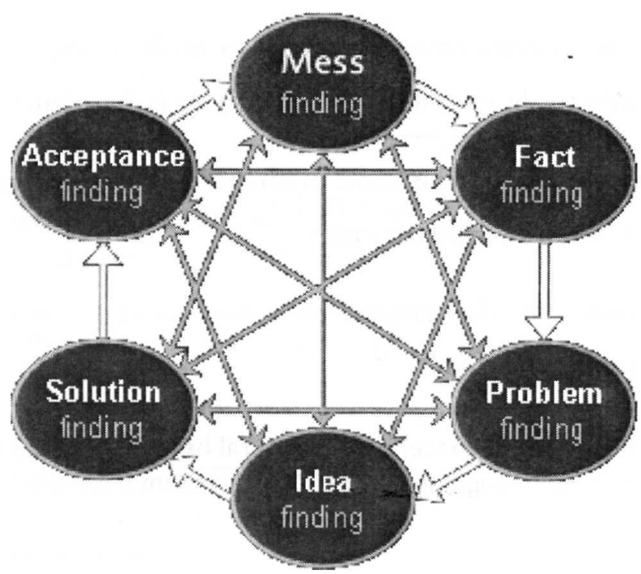

Figure 18: the 6 diamond model shows the different interrelations between CPS parts, together with its cyclic nature.

Depending on the size and complexity of the whole CPS, the process might take a long time and the work group probably will need a facilitator, an expert, or a supervisor to support the different types of decisions to be taken.

Many things are declared in this article, but it totally centered in the inspirational nature of creativity and problem solving. Most of the references are old and it brings no concrete aid to solve problems, but it constitutes a good introduction to the softer view of IPS.

2.3.3. A heuristic-based approach to conceptual design

It is accepted that the highest design standards cannot compensate for poor design decisions taken at the conceptual stage of design which, compared with the rest of the product design, has no available computational tools. Within this hypothesis, Chong proposes an approach to realize a computer-aided conceptual design (CACD) system (Chong et al. 2008).

His most relevant observations are made:

- Conceptual design involves the generation, operation and evolution of a set of relevant knowledge for the problem in hand.
- In practical design problems, the solution space is open-ended
- Explicating all possible concept variants to avoid leaving out potential concepts—is astronomically costly, if at all possible. A strategy is therefore crucial.
- The time available to do so is, more often than not, severely limited due to the common demand for short development cycles.
- Comprehension of the proposition's boundaries is essential, prior to further development or applications

To facilitate the exploration of product concepts, the general best first algorithm (GBF) (Pearl 1984) was adapted, termed as the conceptual design heuristic (CDH) algorithm (Figure 19).

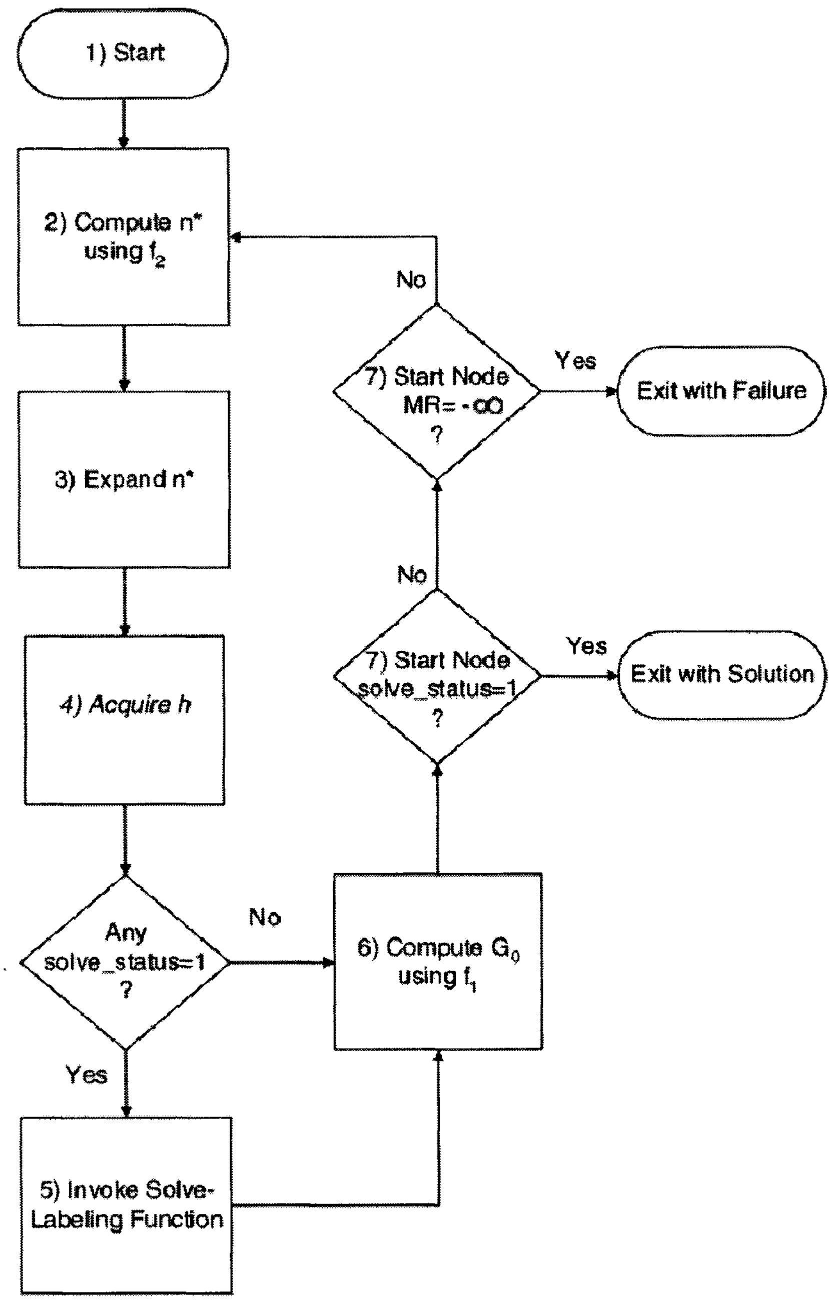

Figure 19: CDH algorithm. For details please refer to (Chong et al. 2008).

Via a single design case study, this product conceptualization approach is demonstrated to strategically guide in the exploration of design concepts, being that its main conclusion: CACD systems should play the role of guiding designers in directions of better solutions when help is needed, by using the key strengths of computers (storage capacity, processing speed and network connectivity).

2.3.4. Genetic algorithms in computer aided design

The article presents the main relations between GAs and CAD, considering that "design is an engineering activity for creating new technical structures characterized by new parameters, aimed at satisfying predefined technical requirements" and that the "design task can often be seen as an optimization problem in which the parameters or the structure describing the best quality design are sought" (Renner & Ekart 2003).

The interest born in that Classical methods (analytical or numerical) for calculating the extrema of a function perform well in many cases of daily practice, but they fail in complex situations with large numbers of parameters and a complicated goal function. The goal function usually has many local extrema, whereas the designer is interested in the global extremum.

Six categories of applications are considered:

- Conceptual design
- Shape optimization
- Data fitting
- Reverse engineering
- Mechanism design
- Robot path design

These are the branches of mechanical engineering in which GAs are most extensively used, all of them with a strong geometric component, present in almost every engineering design problem.

Because the interest is to solve real problems, a key question is how to handling the design constraints. A frequent technique technique is to introduce a penalty into the fitness function: Individuals that do not fulfill the constraints are given penalties proportional to their violation, usually starting with relaxed constraints that get stricter as the GA proceeds. A more reliable approach is to incorporate all constraints into the genetic representation, i.e. to devise a representation which does not allow any individual to violate any constraints. This usually works in simple cases, for more ones it may be very tricky or impossible. Additionally, incorporating too much problem specific knowledge into the representation largely limits the size of the search space and may require the careful definition of specific crossover and mutation operators.

The previous idea of having variable boundaries is interesting as a concept to restrict the search space and orientate solutions, and shares the spirits presented in (Duran-Novoa et al. 2009). The rest of the article gives a good perspective about how GAs are getting into CAD systems, but no "strong" theory is presented, focusing in the process and the main factors to consider.

2.3.5. A new product design method based on integrated intelligence

A lot of hybrid methods, combining case based reasoning (CBR) with a genetic algorithm GA, have been suggested in product designing, having poor performances in practice. To solve that, it is proposed to combine CBR and GA with an algorithm process for simulation of directed similarity association according to human being's thought models (Wen Luo and Chaoan Lai 2009).

To analyze it, the process of creative thinking was divided into three sequential stages:

- Recall
- Association
- Combination

During the recall stage, the divergent thinking can be used by an experienced designer to search heuristically in his memory, to later apply the convergent thinking to find out the most similar cases. In the association stage, the designer's thinking should diverge from the dissatisfaction of the most similar case, looking for some action principles or solutions to improve the most similar case. At last, during the combination stage, designer should integrate his/her strongpoint to find out the most satisfying solution by combining with convergent thinking, and to make a product design scheme. All the three stages compose a whole thinking model of product creative design consisting of alternant converge and divergence. The proposed integrated model of intelligent design system for injection mold is shown in Figure 20.

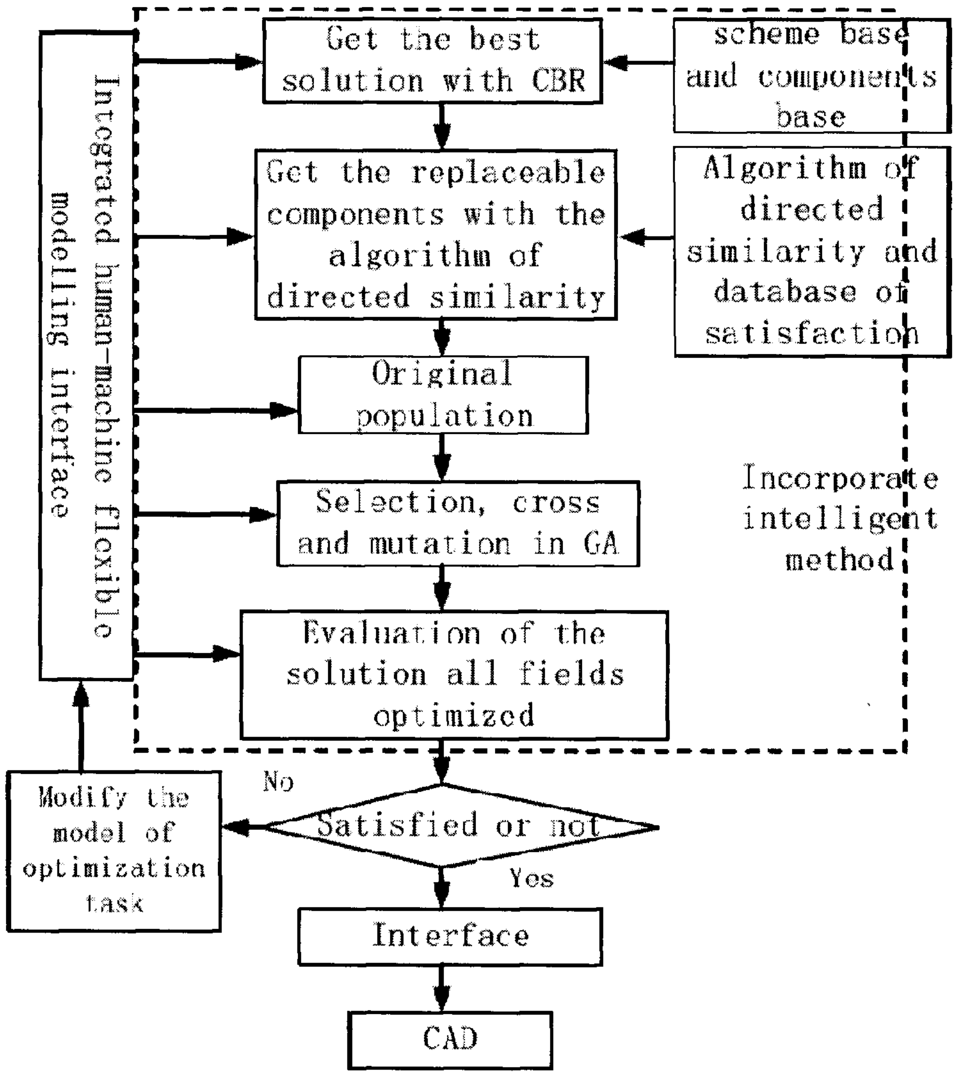

Figure 20: Integrated model of intelligent design system based on creative thinking model (Wen Luo and Chaoan Lai 2009) .

To validate their proposal, a contra-experiment is carried with the design task of injection mold a back mirror frame of a car. In Experiment 1, only the standard GA is utilized, and in experiment 2 the proposed integrated GA (CBR + computation of directed similarity + GA) is used. The results are presented in Figure 21.

Figure 21: Curve of average fitness degree.

It is shown by curve that the average fitness degree in integrated GA with ES is higher than that in standard GA, moreover, the integrated GA converges earlier: "experimental results show that the new hybrid model outperforms other conventional approaches to creative design". Even in the case that the previous is true, several questions arise:

- Which is the real benefit of the hybrid approach?: the first generation has a higher average of 0.3 at the beginning, reduced to 0.1 at generation 200.
- How much it cost?: to have that advantage at the beginning implies a lot more work, with minimum benefits
- How creative design is being measured?: no antecedents are presented.

This study (as many others) claims creativity improvements, but the arguments presented are at least weak.

2.3.6. Virtual genes of manuf. products and their reforms for product innovative design

Based on the idea that the development of manufacturing products is very similar to the evolution of living beings, genetic engineering techniques have been applied by Chen to develop a bionic design theory and methodology for product innovation (K-Z Chen and X-A Feng 2004)

This design method is declared to be totally different from the conventional ones and innovates products via the reform of *virtual genes* of products. Consequently, a virtual gene (or analogous chromosome) of the product is the first stage to fulfil, and it must include both, genetic and evolutionary information. Once the genes are identified, genetic engineering can be applied to reform them based on the mapping network (Figure 22) that shows the way that each gene affects a certain performance of the product, specially the undesirable ones. The product performances in the mapping networks are determined by product designers and/or customers and can be supplemented continuously.

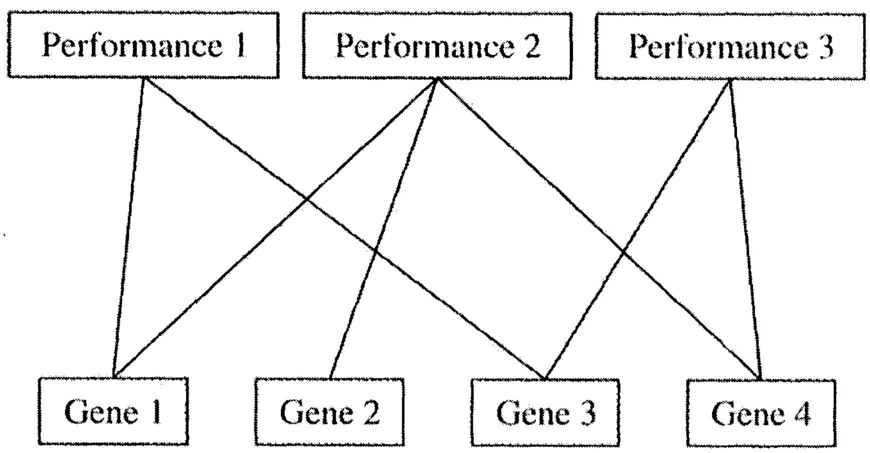

Figure 22: A basic example of a mapping network between virtual genes and performances of a product. For example, performance 1 is affected by both gene 1 and gene 3.

After the gene-performance relations have been established, the product can be innovated by reforming its related genes. The work flow of reforms is shown in Figure 23 and consists of the following steps:

- Step 1. Confirm the defective performances of a product according to its innovative objectives.
- Step 2. Identify defective genes from the mapping networks between genes and performances of a product according to the defective performances confirmed in step 1.

- Step 3. Retrieve superior genes from an evolutionary gene library of the product for replacing the defective genes identified in step 2.
- Step 4. Reconstruct the chromosome by endowing defective genes with zero or much lower probability for being selected as genetic information, grafting the corresponding superior genes to the locations of defective genes and endowing them with higher probabilities.
- Step 5. Evolve the product chromosome using genetic algorithms under a neural-network-based assessment system for performances of the product.
- Step 6. Innovate or reproduce the product according to its evolved virtual chromosome. If the new product does not meet the innovative objectives satisfactorily, return to step 2 to repeat the design procedure until the new product can satisfy the innovative objectives.

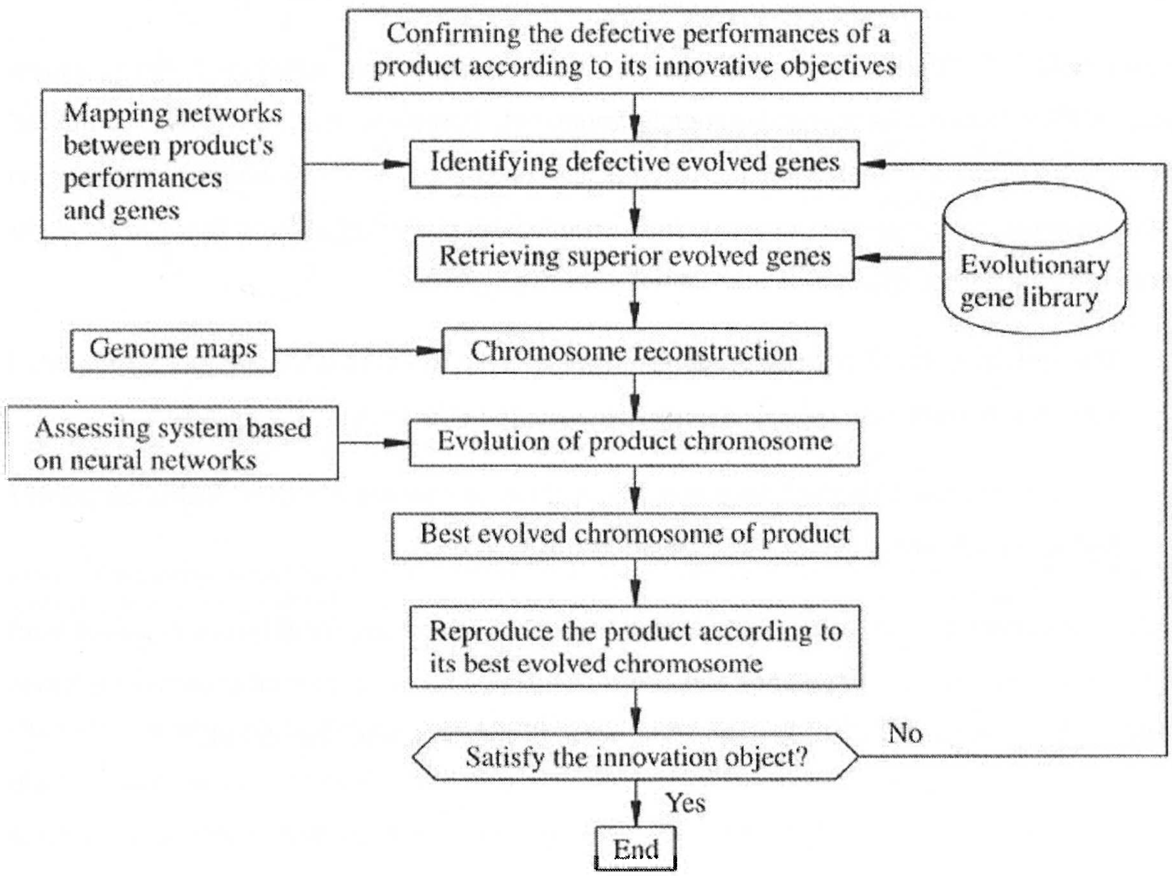

Figure 23: Work flow of the reform of virtual genes

As usual, the problem is automatically assisted after the human performs the most important task: the selection of the performance parameters and the map between genes and defects (in order to reform the genes). Also, the value (relevance) of each link that defines the mapping it is supposed to be done by experts.

2.3.7. Evolving product form designs using parametric shape grammars integrated with GP

In his article, Lee addresses the two critical issues related to product design exploration (Lee & Tang 2009):

- The balance between stylistic consistency and innovation.
- The control of design process under a great diversity of requirements.

The exploration of designs is not only categorized as a problem-solving activity but also as a problem-finding activity, developing a computational framework based on this view. Two computational techniques are utilized: shape grammars (Smyth & Edmonds 2000) and evolutionary computing, which are able to generate "a number of models from scratch with numerical analysis that can be evaluated effectively by the designers". The claimed benefits are:

- The system does all the complicated modeling tasks to construct a number of models from scratch with numerical analysis that can be evaluated effectively by the designers.
- This reduces the designers' time and allows them to concentrate their efforts on performing higher level design activities, like evaluation and decision.

A digital camera design case study is introduced to illustrate the approach (Figure 24). Apart from the two critical issues (stylistic consistency and control of design process), several constraining issues are considered: the configuration, artificial selection, volume, and parametric constraint.

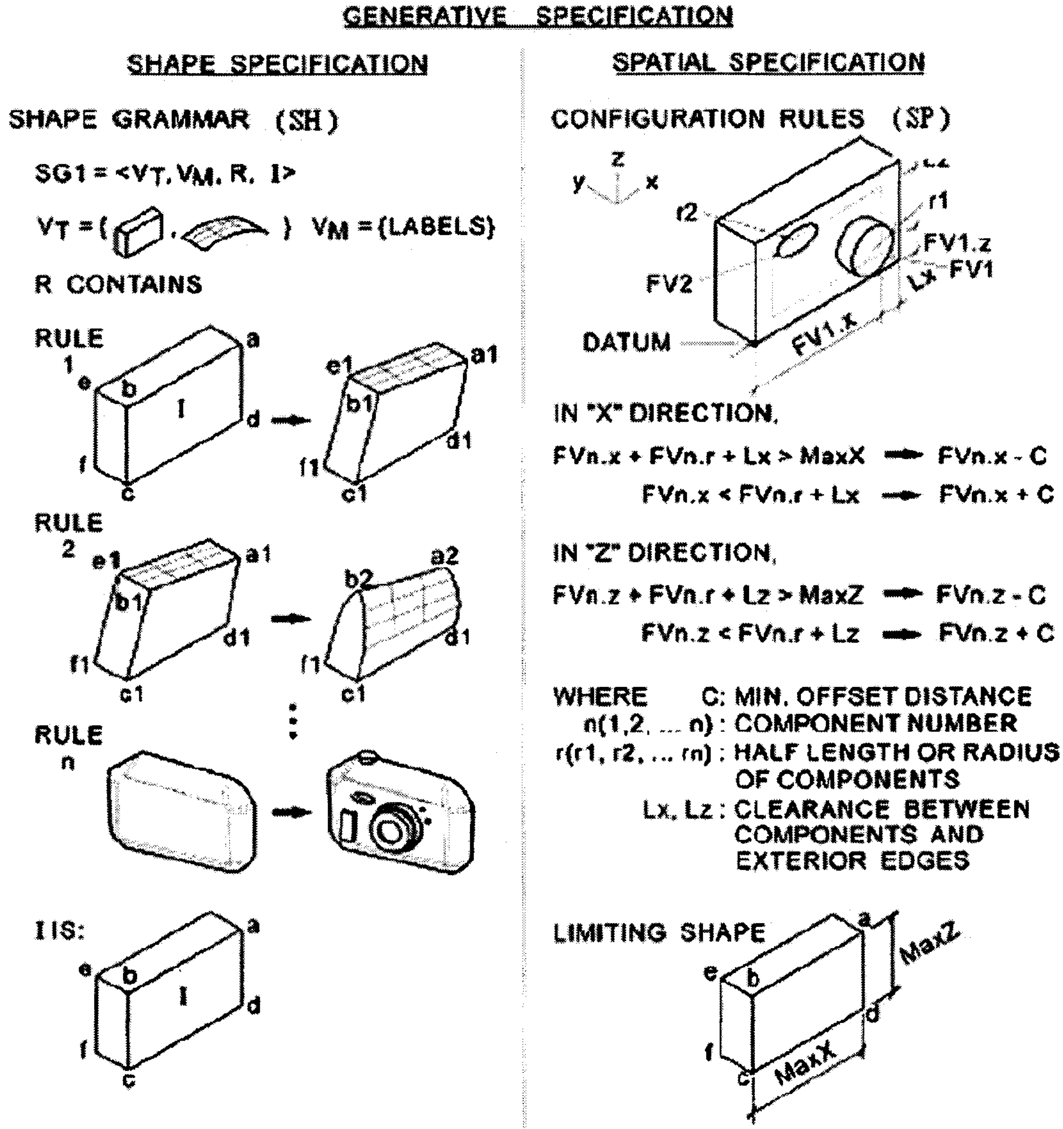

Figure 24: The generative specification of the class of compact camera forms. SH is a specification of a class of shapes and consists of a shape grammar, defining a language of 3-D shapes. SP is a specification of spatial configuration for the shapes defined by SH and consists of a finite list of configuration rules and a limiting shape. (Lee & Tang 2009).

To model the camera, an abstracted core model is composed of combined Non-uniform rational B-spline (NURBS) surfaces and components. With emphasis on the esthetic quality, the exterior of the main body must be a unique design that attracts users, which can be achieved by modifying the control points of each NURBS surface in the core variant model.

In terms of representation issues, under the basic premise that a genetic representation should facilitate the genetic programming to easily manipulate the shape grammar rules, a three-layer representation interface of phenotypes and genotypes called GP-GA-SG is utilized:

- GP: The first layer, is the genotype used by the evolutionary algorithm. The modification variables are the genetic programming components organized as tree structures, being each one represents an evolved program, and consequently a candidate solution.
- GA: is a transformation interface interpreting the effects produced by the genetic programming components, and encoding the shape grammar rules in terms of their rule numbers, associated shape parameters, and constraints.
- SG: is the phenotype used by both the evolutionary algorithm and shape grammars. The SG interface allows mapping between the elements of GA and SG.

In the evolutionary shape grammar-based design system, the genetic programming performs three main functions: modifying alleles within chromosomes using genetic operators, decoding the genotype to produce the phenotype in accordance to the control strategies, and evaluating the phenotype to identify the fittest solutions (Figure 25).

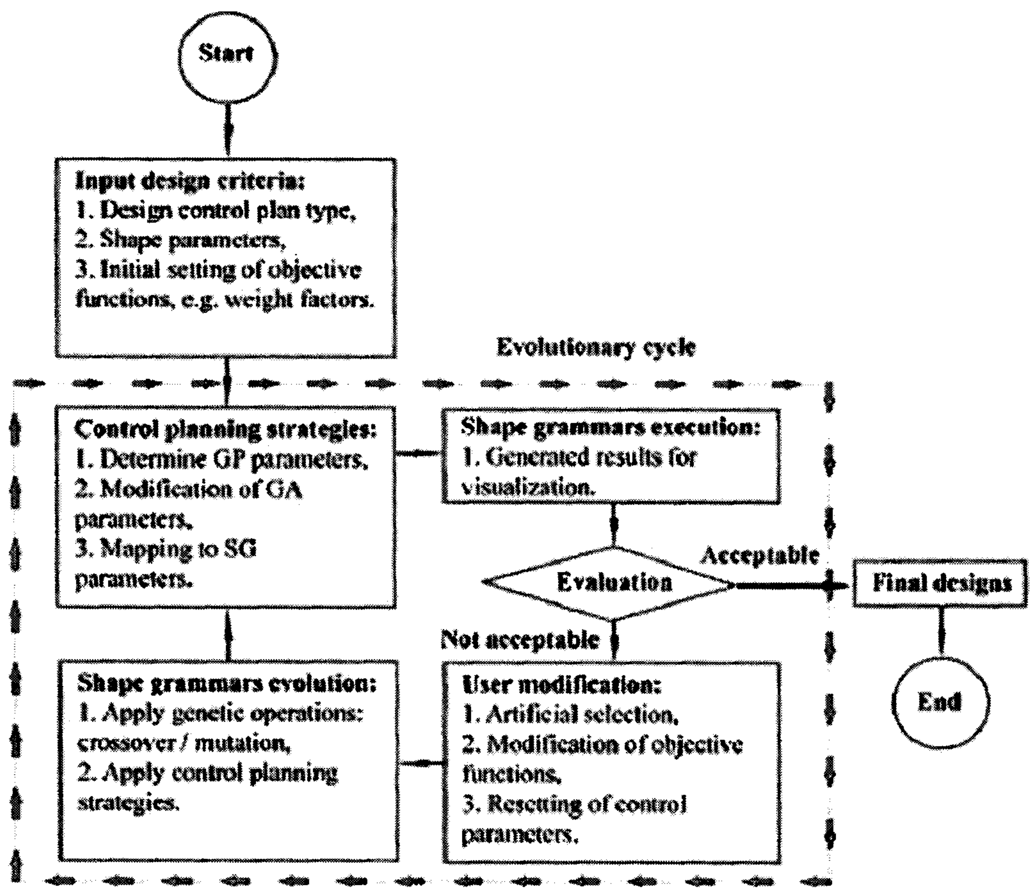

Figure 25: The evolutionary grammar-based design framework.

The article shows how a serious integration of shape grammars with evolutionary computing techniques can be made, and to which extent it facilitates the formulation of design knowledge from the existing designs with parametric shape grammars, doing all the time consuming (not necessarily complicated) modeling tasks to construct a number of models from scratch that can be evaluated effectively by the designers (see Figure 26). The potential of this framework can be further explored in the future in a product design oriented environment that involves complex form generation and configuration optimization.

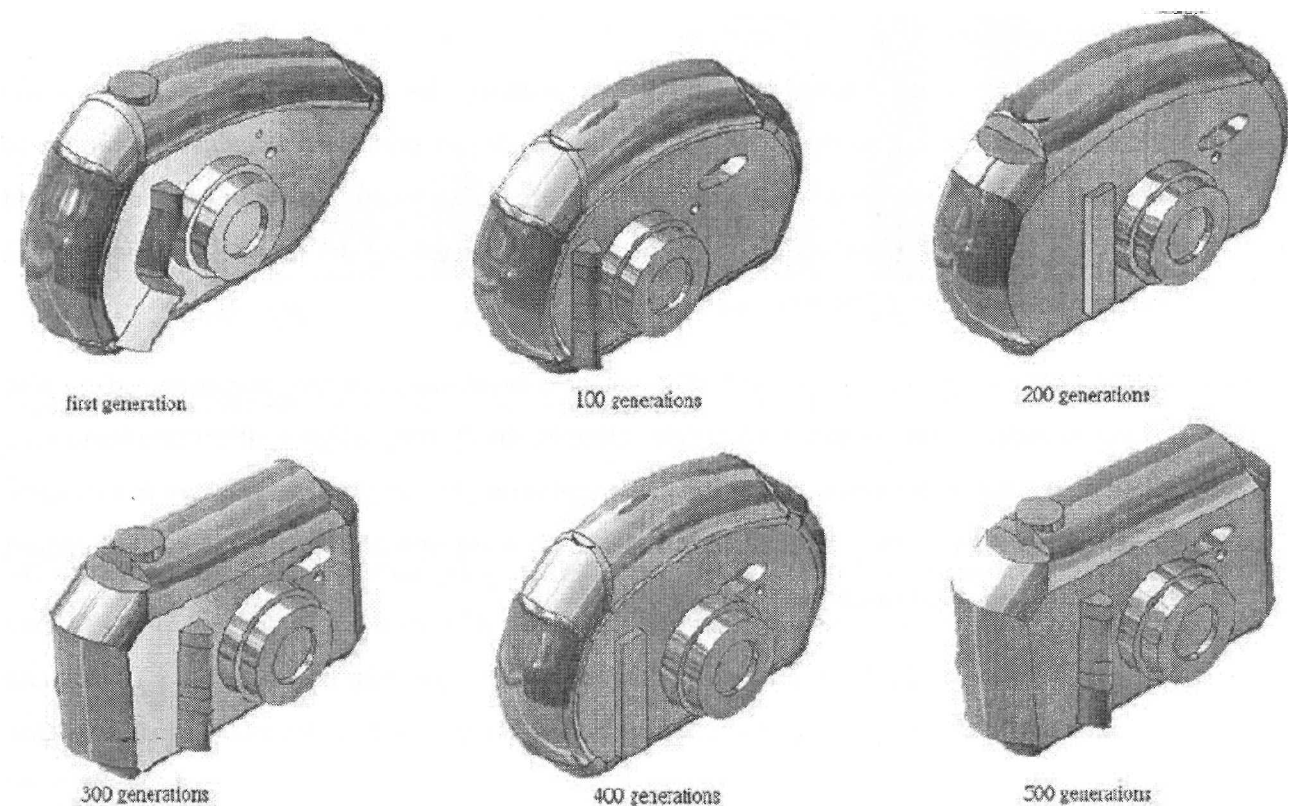

Figure 26: Results obtained from the first generation, 100 generations, 200 generations, 300 generations, 400 generations, and 500 generations.

This way of modeling has several positives aspects (easy to verify manufacturability, material selection). However it can be appreciated the fundamental role played by the human expert in programming and judging, being specially problematic the second one, e.g. which criteria will determine if a design is aesthetically better? . Also, no information about the computational power required is provided.

2.3.8. Innovization: Discovery of Innovative Design Principles through Ev. Optimization

This article proposes a systematic procedure to arrive at a deeper understanding of a multiobjective problem, and not simply to find a single optimal solution (Deb & Srinivasan 2008). Two arguments sustain the previous:

- When a design is to be achieved for the single goal of minimizing-maximizing, usually one optimal solution is the target. When optimized, the optimal solution portrays the design, fixes the dimensions, and conveys little else. A sensitivity analysis could provide some information about the relative importance of the constraints, but it will be valid only close to the single optimum solution.
- Using the optimization tools and subsequent analysis, the insights into the problem will increase, allowing uncovering new and innovative design principles which are common to optimal trade-off solutions. Such commonality principles among multiple solutions should provide a reliable procedure for arriving at a 'recipe' for solving the problem (similar to TRIZ concepts, see section 2.1.9) in an optimal manner.

The example of the design of an electric induction motor is developed, involving armature radius, wire diameter and number of wiring turns as design variables, being the design goal to simultaneously minimize the size of the motor and augment the power delivered. The alternatives will vary from a small motor which deliver only a few horsepower (domestic), to a big one able to deliver a few hundred horsepower (industrial), as presented in Figure 27.

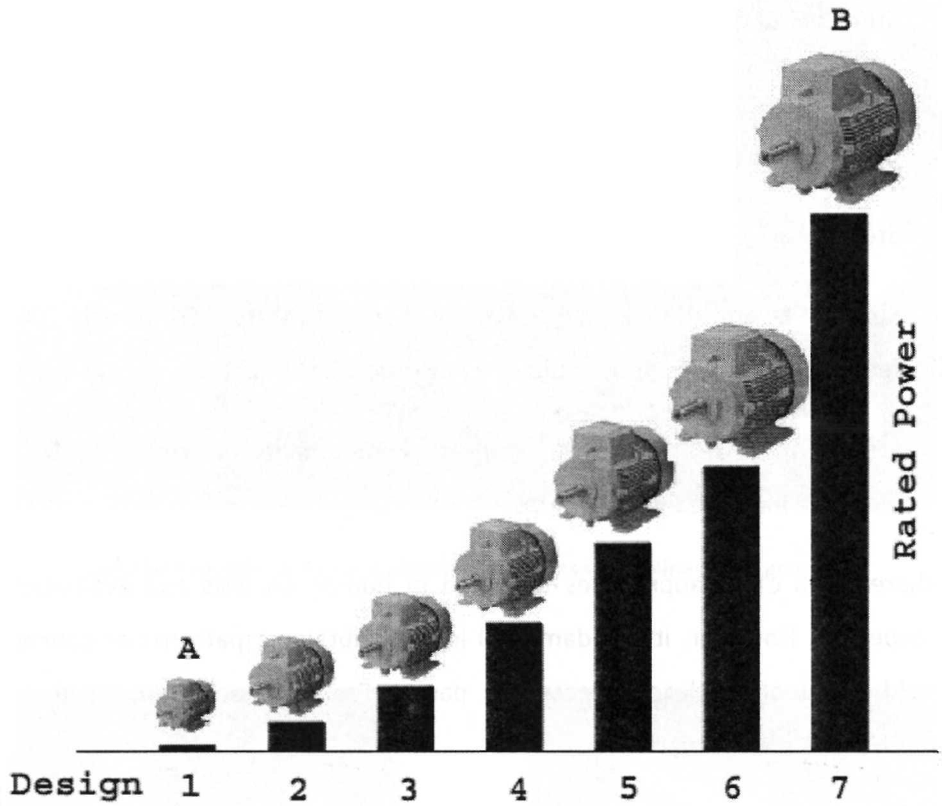

Figure 27: The size and power show a clear conflict when designing a motor of TEFC three-phase squirrel cage induction. The innovization analysis looks if there are any similarities in their designs despite the differences.

After the multiobjective optimization task, there is a set of optimal solutions specifying the design variables and their objective trade-offs. This is the moment to "innovize": perform a post-optimality analysis to investigate if there are some common principles among all or many of these optimal solutions. If the trade-off solutions are really close to the Pareto front, then this analysis should result in worthwhile design.

In the example, to see if there exist any relationship between design variables and objective values exists (like the diameter of the axis is proportional, or the wire is fixed) would be of great importance to a designer. Such information could aid the optimal design of this and any future motor, improving also the theory of motor design, expanding the limitations of the existing procedure allowing new ideas.

The most relevant steps of the suggested procedure are:

- Step 1: Find an individual optimum solution for each of the objectives

- Step 2: Find the optimized multiobjective front by GAs
- Step 3: Normalize all objectives, using ideal and nadir points[5] and cluster a few solutions preferably in the area of interest to the designer or uniformly along the obtained front.
- Step 4: Apply a local search and obtain the modified optimized front.
- Step 5: Perform the "normal constraint method" (Messac & Mattson 2004) starting at a few locations to verify the obtained optimized front
- Step 6: Analyze the solutions for any commonality principles, to be interpreted as plausible innovized relationships.

Innovization shares with other approaches the need of human analysis and evaluation, resorting to intuition and expertise. However, its fundamental idea of obtaining patterns or concepts from static knowledge could eventually aid to surpass the pareto Front, meaning in practice to broke the optimization-innovation (creativity) barrier.

2.3.9. Automatic shape variations for optimization purposes

Performance improvement is commonly obtained through quantitative changes of a parametric design and once optimization reaches its limit, new searches must be carried on through qualitative changes. The author declares that certain innovations can be achieved through methodologies based on optimization processes that focus on functionality and performance where not only design parameters, but also shapes and topologies can be changed (see Figure 29). His research proposes a methodology for automated process integration at mesh level of virtual models, introducing new tools like splines (Figure 28), macros, and TRIZ principles (e.g. segmentation, merging) in order to automate several shape variations that could lead to innovations (Cueva 2006).

[5] Nadir: direction pointing directly below a particular location.

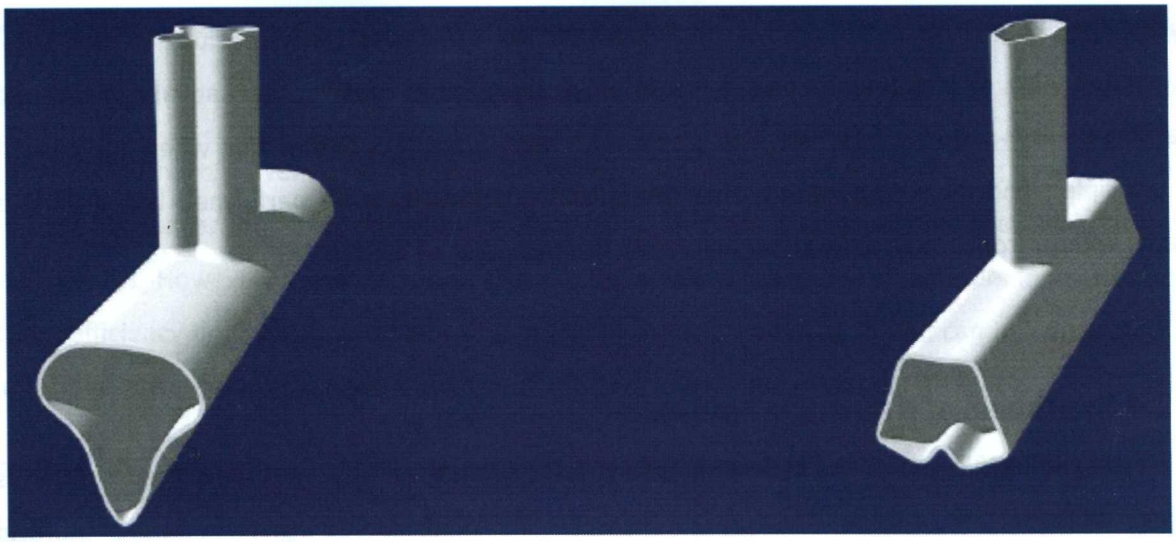

Figure 28: Examples of shape variation possibilities with spline-based geometry

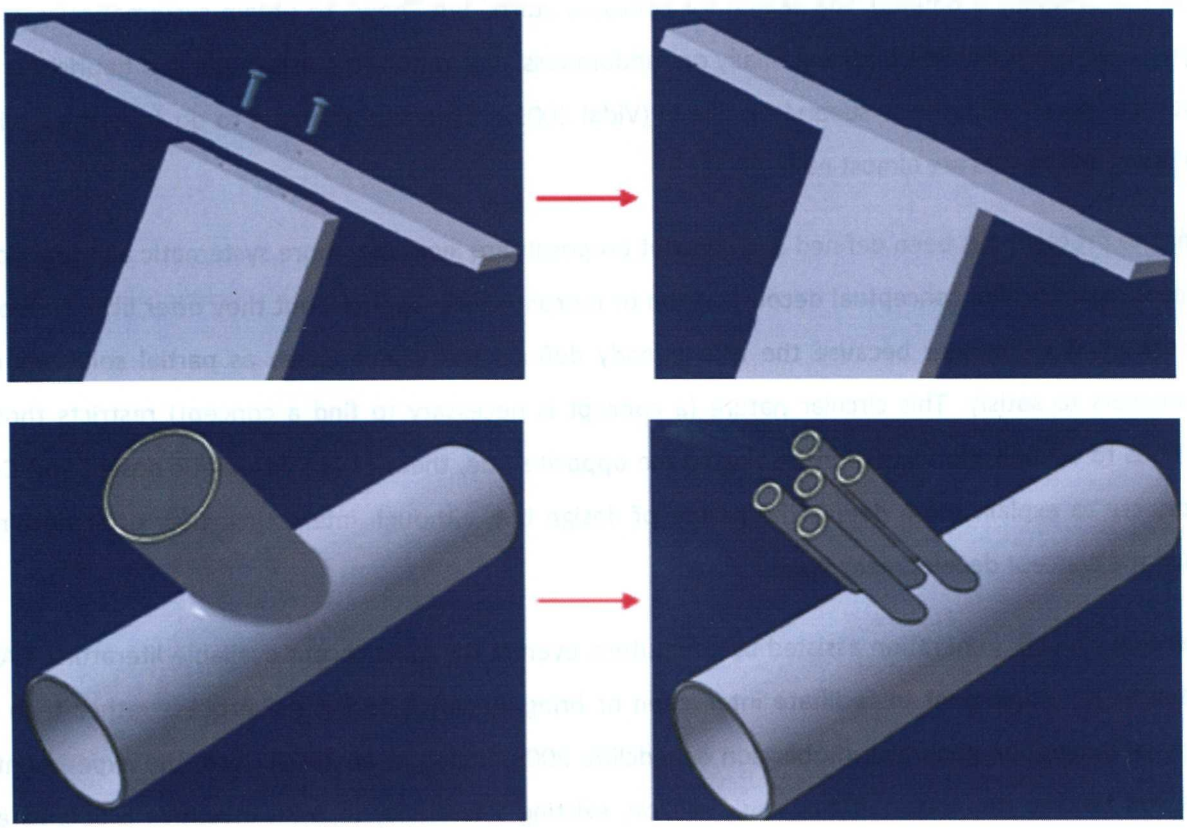

Figure 29: Examples of TRIZ principles interpretations on a CAD context, *merging* (superior) and *segmentation* (inferior). The conditions are determined by the user.

Conceptually, it can be said that this work has practical objectives that are close to the present dissertation. Also, its implementation is a first intuitive approach to consider, very advanced technically regarding its user-software interface. The problem however is the lack of direction when implementing the principles, and more importantly, that the critical activities must be performed based on users' criteria, without conceptual difference when using (for example) a sketch board.

2.4.Synthesis of literature review

The starting point in design is the task clarification. Consequently, an ideal method should aid the user from this definition through the whole process. However, this is considered a given in most of literature, being ignored in more "intuitive" methods and understood as a need in computer assisted ones. TRIZ idea of contradiction is the only concrete proposal to clarify the task.

Once the problem is defined, the search for concepts starts, but "how" to obtain systematically valid alternatives is not validated, relying finally on randomness, inspiration (brainstorming and heuristics) or expertise (via a facilitator or supervisor, like in (Vidal 2009)). Synectics attempts to do something with analogies, but results are almost null.

After the problem has been defined and concept proposals are available, more systematic and practical procedures exist, like conceptual decomposition or morphological analysis, but they offer little support to concept development because the user already defined the search space as partial solutions or parameters to satisfy. This circular nature (a concept is necessary to find a concept) restricts those methods to optimization situations. Going to the opposite side, theories like Axiomatic design and C-K theory try to explain more deeply the nature of design but, although interesting, they offer nothing during the concept development stage.

Regarding concept generation assisted by computers, even in the most recent available literature, CAD is seen as a complement to facilitate interaction or bring guidance during the process rather than a potential innovation generator (Robertson & Radcliffe 2009; Chong et al. 2008). Also, the experiments observed tend to mix tools without any direction, existing a tendency to treat inventive problems as optimization ones, assisting IPS computationally after the most critical operations are performed by humans, like determine variables, mapping causes&effects, and do evaluations (Wen Luo and Chaoan Lai 2009; Renner & Ekart 2003). In other words, despite the good intentions, high level operations are

Conceptually, it can be said that this work has practical objectives that are close to the present dissertation. Also, its implementation is a first intuitive approach to consider, very advanced technically regarding its user-software interface. The problem however is the lack of direction when implementing the principles, and more importantly, that the critical activities must be performed based on users' criteria, without conceptual difference when using (for example) a sketch board.

2.4.Synthesis of literature review

The starting point in design is the task clarification. Consequently, an ideal method should aid the user from this definition through the whole process. However, this is considered a given in most of literature, being ignored in more "intuitive" methods and understood as a need in computer assisted ones. TRIZ idea of contradiction is the only concrete proposal to clarify the task.

Once the problem is defined, the search for concepts starts, but "how" to obtain systematically valid alternatives is not validated, relying finally on randomness, inspiration (brainstorming and heuristics) or expertise (via a facilitator or supervisor, like in (Vidal 2009)). Synectics attempts to do something with analogies, but results are almost null.

After the problem has been defined and concept proposals are available, more systematic and practical procedures exist, like conceptual decomposition or morphological analysis, but they offer little support to concept development because the user already defined the search space as partial solutions or parameters to satisfy. This circular nature (a concept is necessary to find a concept) restricts those methods to optimization situations. Going to the opposite side, theories like Axiomatic design and C-K theory try to explain more deeply the nature of design but, although interesting, they offer nothing during the concept development stage.

Regarding concept generation assisted by computers, even in the most recent available literature, CAD is seen as a complement to facilitate interaction or bring guidance during the process rather than a potential innovation generator (Robertson & Radcliffe 2009; Chong et al. 2008). Also, the experiments observed tend to mix tools without any direction, existing a tendency to treat inventive problems as optimization ones, assisting IPS computationally after the most critical operations are performed by humans, like determine variables, mapping causes&effects, and do evaluations (Wen Luo and Chaoan Lai 2009; Renner & Ekart 2003). In other words, despite the good intentions, high level operations are still human domain, as presented also in (K-Z Chen and X-A Feng 2004; Lee & Tang 2009; Deb & Srinivasan 2008).

In this area EAs have proven to be efficient converging to optimal values and providing novel ideas when searching big spaces, but they are not free from being blocked at Pareto fronts that avoid achieving "ideal" results. However, the observation that adding a few unnecessary parameters in an attempt to ensure sufficiency does not tend to slow down EAs overmuch (Poli et al. 2008), can be considered as a countermeasure whenever is necessary to work modifying (expanding) the search space.

In resume, the most promising ideas to consider developing concepts without relying in expertise and intuition are TRIZ contradiction (to define the problematic) and evolutionary assistance (to develop concepts).

3. Dialectic Model

3.1. Introduction

In this chapter it is developed the model that aims to answer the questions presented in section 1.2:

- Which are the problems when solving problems?
- How actual methods are dealing with those problems, including the user?
- How to make a model as user independent as possible, increasing the automation of IPS in order to utilize CS to actively aid the concept generation?

The main theory utilized to guide the answer search is *dialectics*. Since dialectics explains how the changes operate, it is believed that its principles can be used to model IPS and, based on them, to increase the automation of the concept development. In other words, the main hypothesis is that "dialectics" is an adequate framework to develop a more user independent IPS method.

3.2. Analysis of the IPS problematic

In this section, the relations between the different conflicts during the *concept generation* are presented and analyzed. In parallel, the requirements that a method should fulfill to be considered useful became explicit.

3.2.1. Activities Performed

In order to determine the general problems of IPS, several techniques were utilized. This process raised new questions and focused the previous efforts, clarifying also how the different methods deal with each problematic. Figure 30 shows the areas where the analysis is concentrated.

3. Dialectic Model

3.1. Introduction

In this chapter it is developed the model that aims to answer the questions presented in section 1.2:

- Which are the problems when solving problems?
- How actual methods are dealing with those problems, including the user?
- How to make a model as user independent as possible, increasing the automation of IPS in order to utilize CS to actively aid the concept generation?

The main theory utilized to guide the answer search is *dialectics*. Since dialectics explains how the changes operate, it is believed that its principles can be used to model IPS and, based on them, to increase the automation of the concept development. In other words, the main hypothesis is that "dialectics" is an adequate framework to develop a more user independent IPS method.

3.2. Analysis of the IPS problematic

In this section, the relations between the different conflicts during the *concept generation* are presented and analyzed. In parallel, the requirements that a method should fulfill to be considered useful became explicit.

3.2.1. Activities Performed

In order to determine the general problems of IPS, several techniques were utilized. This process raised new questions and focused the previous efforts, clarifying also how the different methods deal with each problematic. Figure 30 shows the areas where the analysis is concentrated.

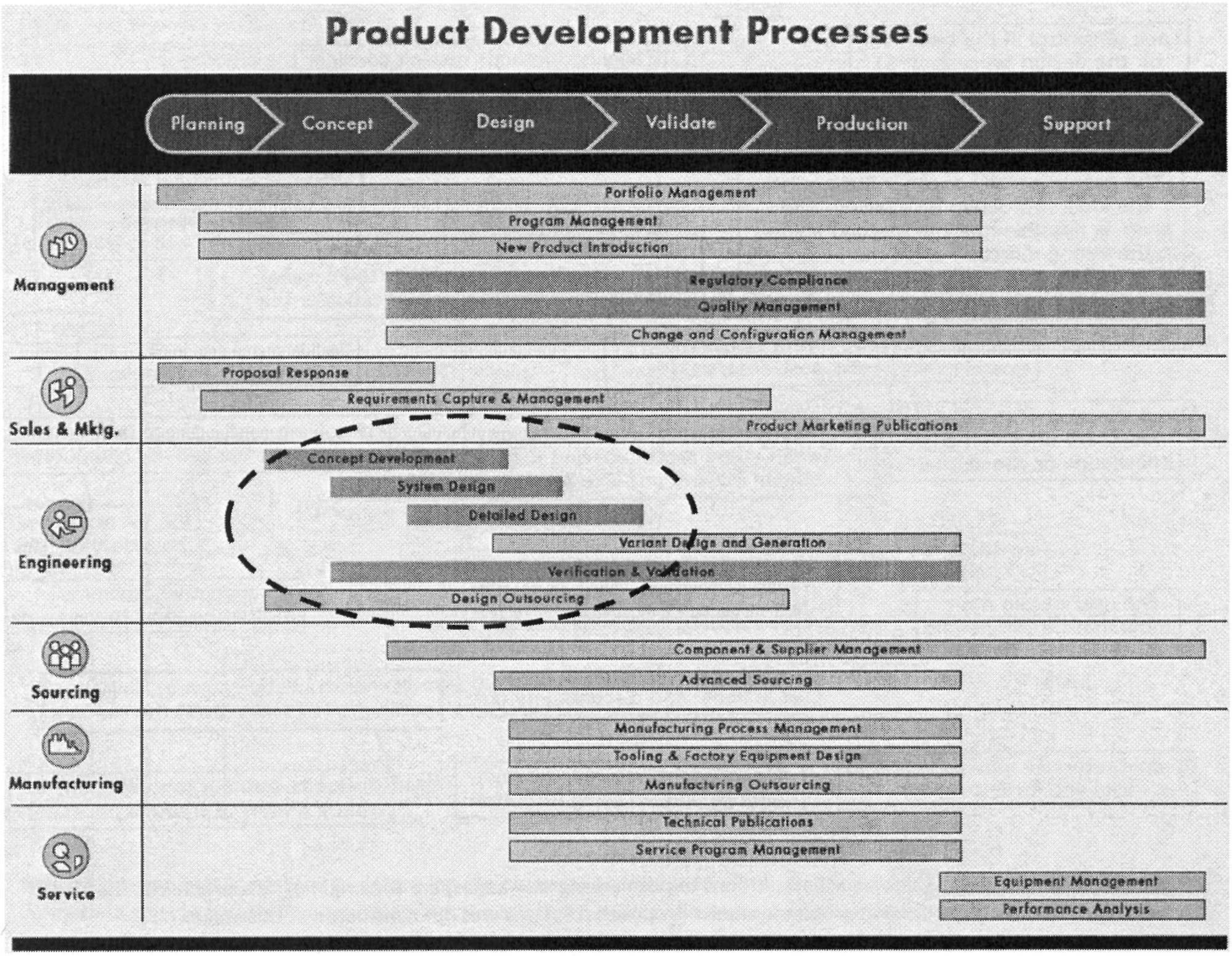

Figure 30: Generic product development process. The oval shows the main area of interest (source: PTC, http://www.ptc.com/WCMS/files/56916/en/3549_Mathcad_bro_ViewONLY.pdf).

Within this context, a preliminary *Network of Problems* (NoP) (Khomenko & De Guio 2007) was developed obtain a general picture (Figure 31).

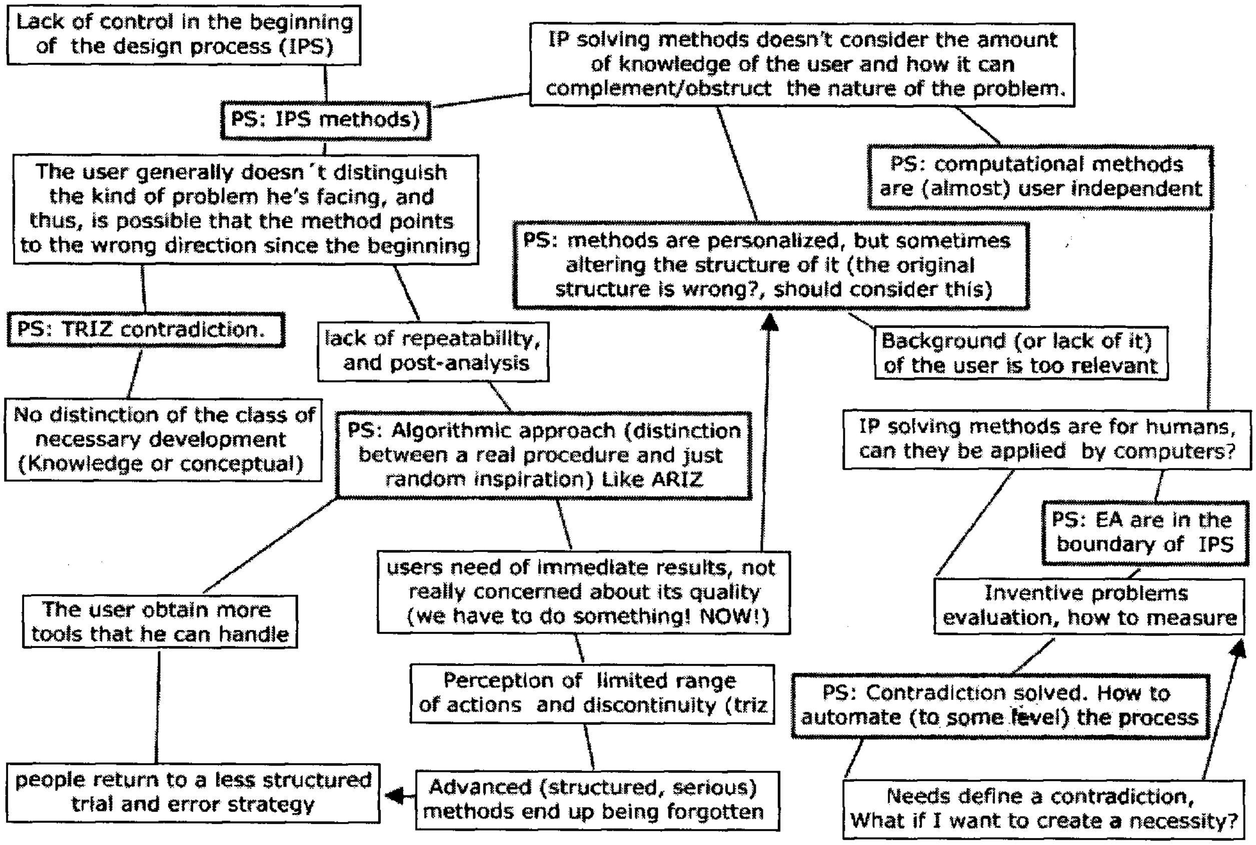

Figure 31: Preliminary NoP. On it, partial solutions to problems (green boxes) are the origin of new problems. The kind of relation between problems and its proposed solutions was not considered at this stage.

To clarify the potential areas of contribution and how to increase the control of the IPS process in general, a second diagram was made (Figure 32), having in mind the obstacles in the utilization of IPS methods.

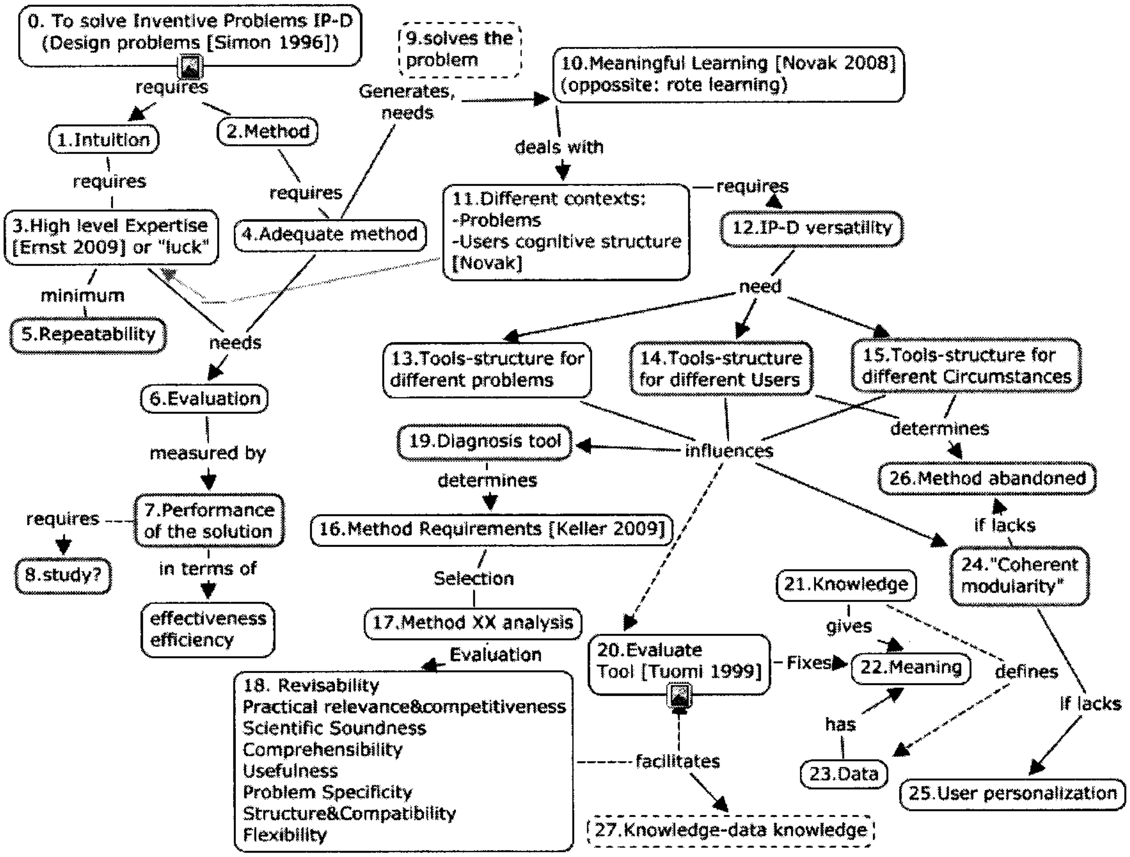

Figure 32: Second NoP, oriented to determine the obstacles to utilize IPS methods. The necessary characteristics of an IPS method start to be clarified, together with the relation between those characteristics and the major consequences.

During the development of the NoP's, several fundamental questions arise:

- Why it could be useful (the automation of IPS)?
- Which problem is solved if IPS is performed "automatically"?
- What are the characteristics of the output (method, software, other)?

In relation to this particular research, the second one is the most critical: *Which problem is solved if IPS is performed automatically?*. The answer is **the unpredictability of the IPS process and its dependence of specific individuals** ("inspiration"). There are terms that need a more precise definition within the context (unpredictability, automatically), but both the question and the answer clarify the needed development.

Tentative answers to the other posed questions are:

- Why it could be useful (the automation of IPS)? : Actual IPS methods in most of the situations are not used, used superficially, or abandoned.
- What are the characteristics of the output (method, software, other, don't know)? : The first output is a method (or a contribution to an existing one) with a clearer path towards its computational implementation.

In order to register the problems that IPS methods should overcome, a first list of desirable characteristics was developed, which is presented in. This was done arbitrarily to have a reference point, since in literature the analysis is concentrated in what the methods do ("state the problem", "define a technology") without explaining the details about how to get there (like in (Eder 2009)). The relative relevance of the listed characteristics (left column) is given by the problems that emerge if they are not fulfilled (right column). For example, L1 and L2 mention that the method should be able to be learned and used gradually, because if don't it is probable that the user will avoid it or utilize it superficially, becoming an obstacle that delivers no results (R1). L7 and L8 point towards the same problem (R4) but its relation is less evident.

Table 4: The list of desirable characteristics of a method (left) is matched with the consequences if they are not considered. Both columns do not attempt to be exhaustive, but a reference point to start the cycle of analysis and improvement regarding the cause

List of desirable characteristics	Results when not considered
L1. It can be learned/used gradually L2. The user doesn't avoid it	R1. If the intended user avoids the method (or cannot learn it), no result is possible.
L3. Allows transforming tacit knowledge into explicit one. L4. Its process is repeatable and traceable, user independent	R2. The "individual" becomes critic, no substantial difference with experience-intuition.
L5. It gives a quick and valid picture of the situation L6. Input/output (of the method) are well defined	R3. The path towards the solution is diluted, so the search becomes random with unclear objectives.
L7. In dead ends, it shows the potential reasons in order to discover bad assumptions L8. Allows to see things from different perspectives	R4. Circular reasoning, denial, prejudges and others could block the solution.
L9. It takes the user to the "same place" cheaper or faster. L10. Learning time < Saved time, inversion<<benefits.	R5. If a method isn't competitive (and cost/benefit viable) mustn't be used.
L11. Allows to quickly rejecting unrealistic proposals. L12. Allows to rank multiple problems	R6. Higher probabilities of selecting a wrong problem(s) (with less impact, less efficient, more difficulties...).
L13. Context restrictions can be included (time, knowledge, money). L14. Other area observations can be included	R7. The method turns into an exercise, with almost no real chances to be used.
L14. Other area observations can be included L15. Versatility: Adaptable to different problems, users and circumstances	R8. The users will be limited, and thus the comprehension and collaboration (only experts?).

To explore and validate this first arbitrary list of requirements, the *Concept Generator* (section 2.1.8) was checked, paying special attention to its efficiency & efficacy. After this analysis, the problems were contracted and related to different generic characteristics of IPS methods:

- How the method is presented to the user, how it has to be used (M1 Presentation)
- Which requires, from the user-system, to perform (M2 Requirements)
- On which conditions operates (M3 Restrictions)

- Expectable results from its utilization (M4 Results)

This is presented in Figure 33, and it was done to clarify which part of the method can tell if the problem has been first considered and how it proposes to solve it, verifying also if an evident problematic was left out, implying the incompleteness of the list of desirable characteristics.

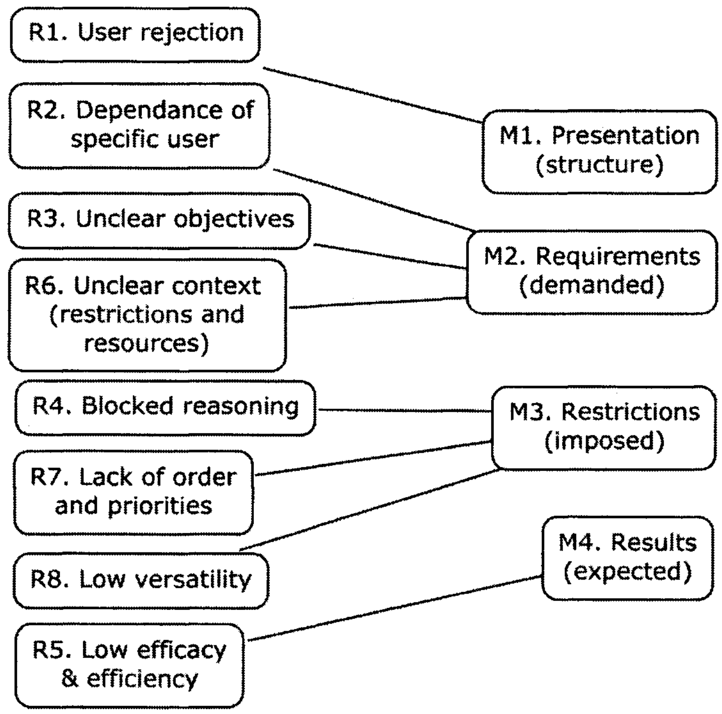

Figure 33: Potential problems of IPs methods and its most related areas. The codes of the left column (Rx) are in correspondence with Table 4.

Finally, to put in perspective the user and objectives, TRIZ's *system operator* (Altshuller 1984) was utilized. Since the intended future practical application of this research is the incorporation of computational technologies in the IPS, it was developed a 20 years gap diagram, listing the relevant changes about how the computational tools available assist the components of an IPS method at each stage. This is presented and explained in section 3.2.2.1.

With the context clearer, a new NoP was developed (section 3.2.2.2) and later expanded, presenting its problems-components as physical and technical contradictions (section 3.2.2.3).

3.2.2. Diagrams and descriptions

3.2.2.1. System operator

The product development process (PDP) for a "standard product" can be completely automatic, e.g. via internet the client declares the parameters that he need and this generates a CAD model, which can be simulated, validated mechanically, and manufactured. It can be almost human independent. But the situation is completely different with IP. No situation that implies to overcome an unknown obstacle can be handled by an automatic system, and thus, product development as a whole cannot become a standard procedure.

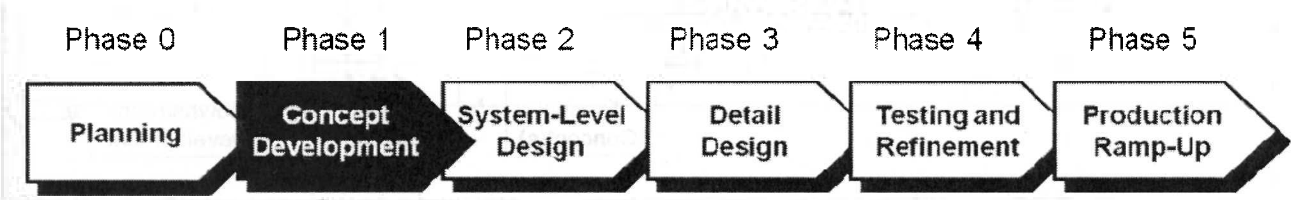

Figure 34: standard Product Development Process (Ulrich & Eppinger 2003) . The overall automation levels are dependant of phase one.

What is done in practice is that a professional in the field of the specific product needs to receive the problem and then develop and evaluate an adequate proposal. During the concept development stage of the PDP globally an *expert* is needed, because of the concept generation and evaluation: a competent user has not the necessary characteristics to make appropriates decision, despite the fact that he can state the problem and embody it.

Problem statement	→ **Concept development** →	**Embodiment**
Competent developer	Expert developer	Competent developer

This dependence of high level users reduces the control and predictability of the product design as a whole, and could become a "bottleneck" in the PDP. Also experts need long training, are few, and they are expensive.

The previous also implies that any development of CS in other stages of the PDP will not provide a reliable improvement until the concept development stage has been balanced in the solution of IP, specifically in the specification, generation, and selection of concepts ("a chain is only as strong as its weakest link").

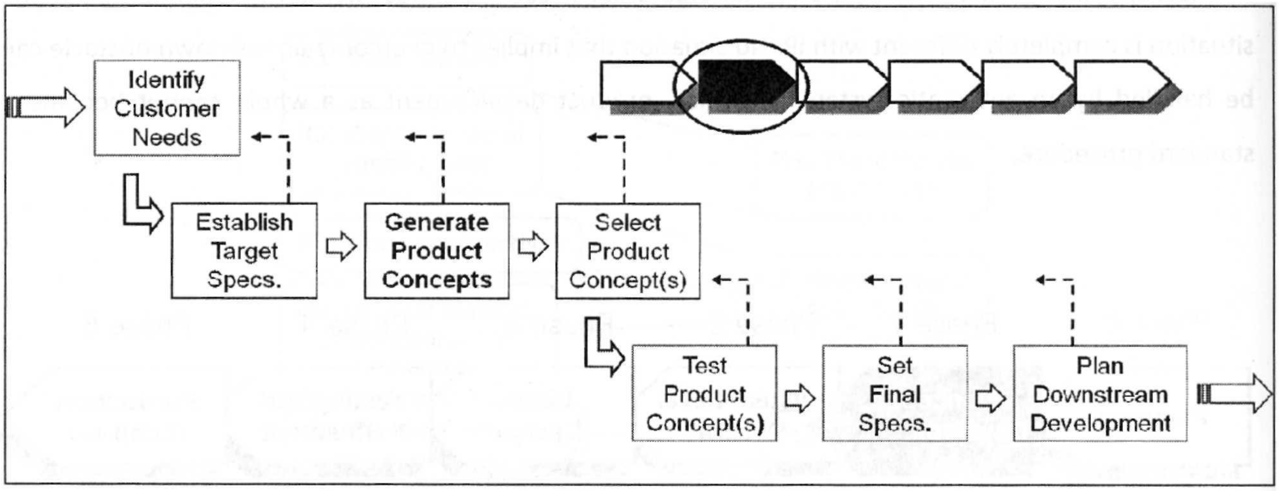

Figure 35: PDP sub stages. The concept generation is the area to focus.(Adapted from (Ulrich & Eppinger 2003))

Figure 34 and Figure 35 show the most relevant relations in the studied system. To visualize the situation regarding how CS has actually supported concept development, TRIZ *system operator*[6] technique was used. The super-system and sub-system are part of this analysis, which looks at the past, present, and future state of each level, in this case utilizing a 20 year gap. This gap was chosen because during the nineties internet and computers began to enter every aspect of life, including design: more powerful computers were able to model (instead of drawing) and knowledge turned more accessible. A simplified version is presented in Table 5.

[6] Explained in (Altshuller 1984), and in several internet sites (like Wikipedia or the Triz-journal). A.k.a. "9 windows".

Table 5: General view of the system operator. It was constructed under the look of how CS available at the time assists (expected to) the user at each level and time.

	Past (1991)	Present (2011)	Future (2031)
Super System: Concept development	• Bibliography	• Bibliography • Info collection • Guidelines	• Simultaneity
System: Concept generation	• Databases	• Suggestions	• Global coherence • Coordination
Sub System: System components	• Modeling	• Modeling • Modeling/Edition • Simulation	• Real time validation • Alternatives availability

The Sub-system is an oriented adaptation of *the five step concept generation method* (Ulrich & Eppinger 2003), which consist in:

- Step 1, clarify the problem: decompose a problem into simpler parts, and focus initial efforts on the critical ones.
- Step 2, search externally: interview lead users, consult experts, search patents, published literature, benchmark related products.
- Step 3, search internally: both, individual and group sessions, can deliver useful hints for generating concepts.
- Step 4, explore systematically: using the classification tree or a combination table to managing the exploration process.
- Step 5, reflect on the solutions and the process: feedback.

The proposed steps are too generic to be used, especially in this evaluation focused on computer science assistance. To overcome that, the Sub-system stages utilized are four:

- Problem clarification: how CS aid the clarification of a problem (step 1)
- Search: how CS aid the search and documentation of alternatives (steps 1 and 2)
- Analysis: how CS aid to determine the viability of an alternative (step 4)
- Proposal: how CS aid the selection and reutilization of an alternative (step 5)

Figure 36,Figure 37 and Figure 38 presents each cell of Table 5 in detail, followed by brief explanations. Each color indicates a different level (orange= super; cyan= sub). Several acronyms are utilized, whose meanings are presented in Table 6.

Table 6: Acronyms utilized in the system operator (Figure 36, Figure 37 and Figure 38).

Acronyms on System Operator	Sub-system components
• SU#: super system level • SY#: system level • SB#: sub system level • - : past, +: future	• U: User needed expertise[7] • T: Techniques & Tools • C: Circumstances (external-explicit) • A: Assumptions (internal-implicit) • E: Evaluation.

The starting point of the analysis is the present of concept generation (SY1), which is a part of Concept development (SU1) and the origin of all sub components (SB#). In general, it can be observed that in the past CS bring order and speed to *Search*, and many tools to assist *Analysis* and *Proposal*. However, an expert was needed to take advantage of those developments, and there was a serious issue about synchrony.

[7] Novice, advance beginner, competent, proficient, expert, master, and visionary. The categories are taken from (Eder 2009)

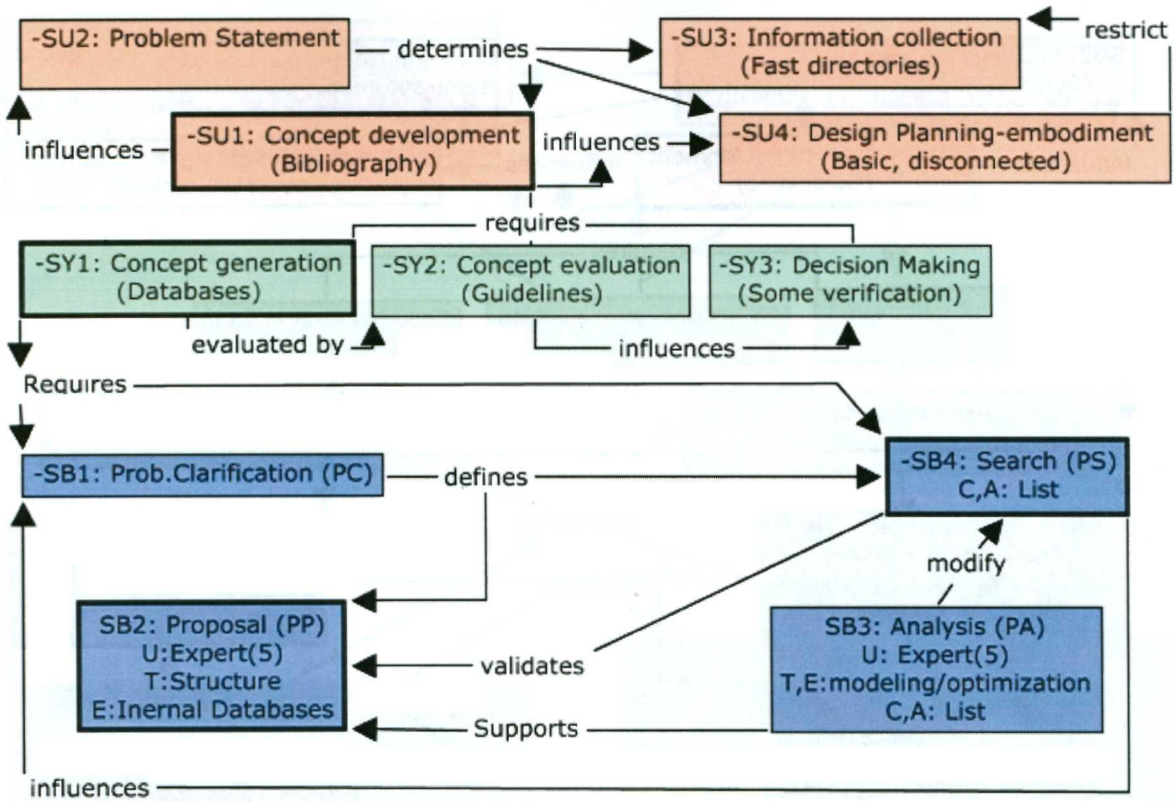

Figure 36: System operator, 1991. CS participation in Concept generation was starting, mainly as a way to manage big amounts of information.

At the present, CS has augmented the level of automation in almost every stage of the concept development, making to descend the needed user expertise, especially in the most technical areas. Many elements of the product design chain that were disconnected has been linked (like CAD-CAM) , and now there exist real time support in the *Problem clarification* stage (absent in the past) through interactive databases and semantic search (like in Goldfire innovator[8]) that can even make suggestions of valid solutions.

However, the *Proposal* stage has not been assisted by the CS, which has focused on the optimization instead. Talent, creativity, and inspiration are (still) some of the adjectives used to describe what is needed, despite the work being doing to control this design stage (some of it presented in the literature review).

[8] http://www.invention-machine.com/

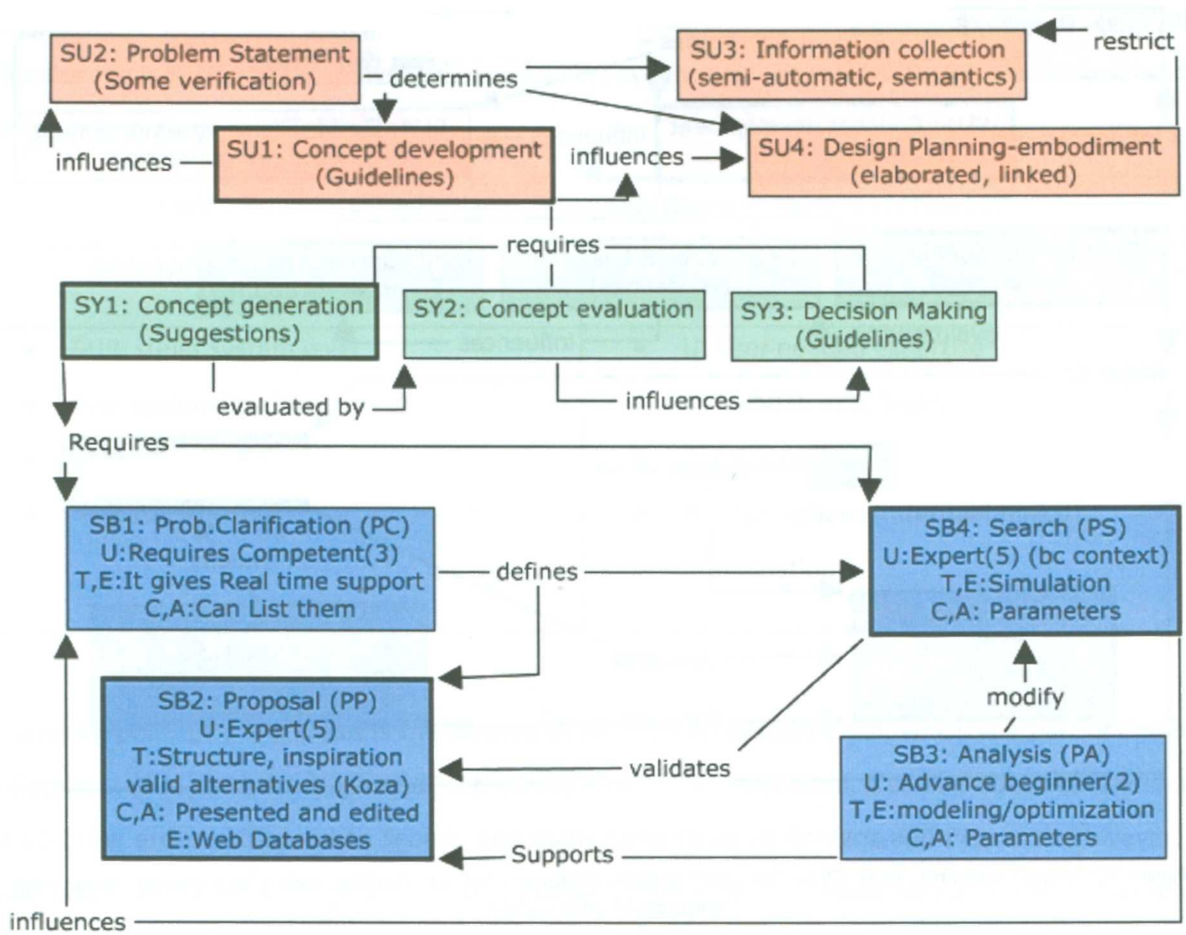

Figure 37: System operator, 2011.

Following the trend towards integration observed from past to present, it is expected that the tools that today are linked will be combined, giving a constant feedback along the Concept development process. As a result of that integration, in the higher level there will be possible to distinguish just two stages:

- Definition & Research: The statement and research of the problem will be simultaneous using natural language.
- Proposals & Viability: every proposal will be in constant evaluation, considering external databases (like patents) and internal ones.

Something similar will occur at system and sub-system level, where additionally the relations between them increases and tend to become bidirectional due to the reduced length of the feedback loop.

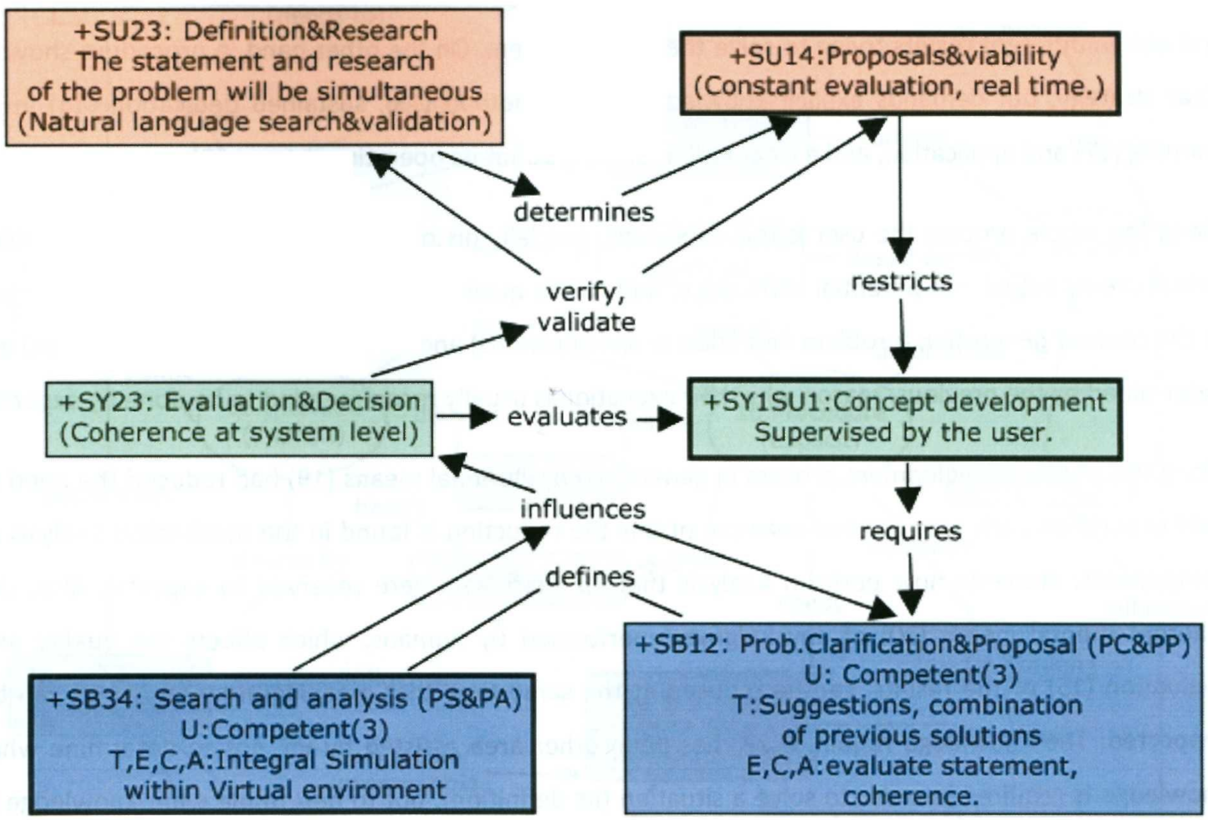

Figure 38: System operator, 2031.

To reach that level of integration, it is necessary the ability to generate concepts in a more reliable and predictable manner, pointing towards **the idealized idea** of "automatically generate valid solutions to inventive problems". This will also solve the problem of lowering down the necessary global expertise level of the user (from expert to competent) of the product design as a whole. To attempt to reduce even further the user level is useless, **unless the problem formulation and restrictions** can be stated by an "artificial intelligence", setting a new lower limit to the *product development* process as a whole.

3.2.2.2. Network of problems (NoP)

In Figure 39 it is presented the NoP during the *concept generation*. On it, the starting point is the user (01), who needs to recognize that he's facing a problem (02). He also needs to have, at least, an intuitive idea of what results he is expecting (03) and the strategy to achieve them (11).

To solve problems, the user can utilize two generic methods: intuition or a formal procedure (4, 8, and 10). Intuition can give quick results (09), but it's too dependent of user's capacities (16) and his implicit knowledge (7), having also almost no repeatability: even the same person can follow a different path and obtain different results trying to solve the same problem. On the other hand, a procedure shows a clear strategy, but demands explicit knowledge of the method (13), sustained dedication (14) in its learning (17) and application, and a clear understanding about its operational range (5).

Along the whole process the user is too important, specially his perception and capacities (16), which determine its actual and potential knowledge, and so the quality (18) of candidate solutions. The input of the *concept generation* (problem definition or specifications) and its output (valid design concept) are determined by the previous, reason why their execution is usually recommended to be done by experts.

Along the product development process in general, computational means (19) had reduced the need of field experts on each area (a good example of this the reduction is found in the mechanical analysis of components: students now perform analysis that 10 years ago were reserved to experts). Also, the inherent mental inertia (20) of any judgment performed by humans, which affects the quality and evaluation (15) of the results, can be reduced at the same time that meaningful learning (17) can be supported. The knowledge required (13) has been other area assisted by CS, not to determine what knowledge is required in order to solve a situation (its definition), but to determine what knowledge is required to execute a concept proposal (its implementation) and make also this knowledge available at the needed time.

This second NoP puts on perspective the macro situations that IPS methods have to deal with, and the reasons why a method that doesn't considers them has a minimal practical convenience.

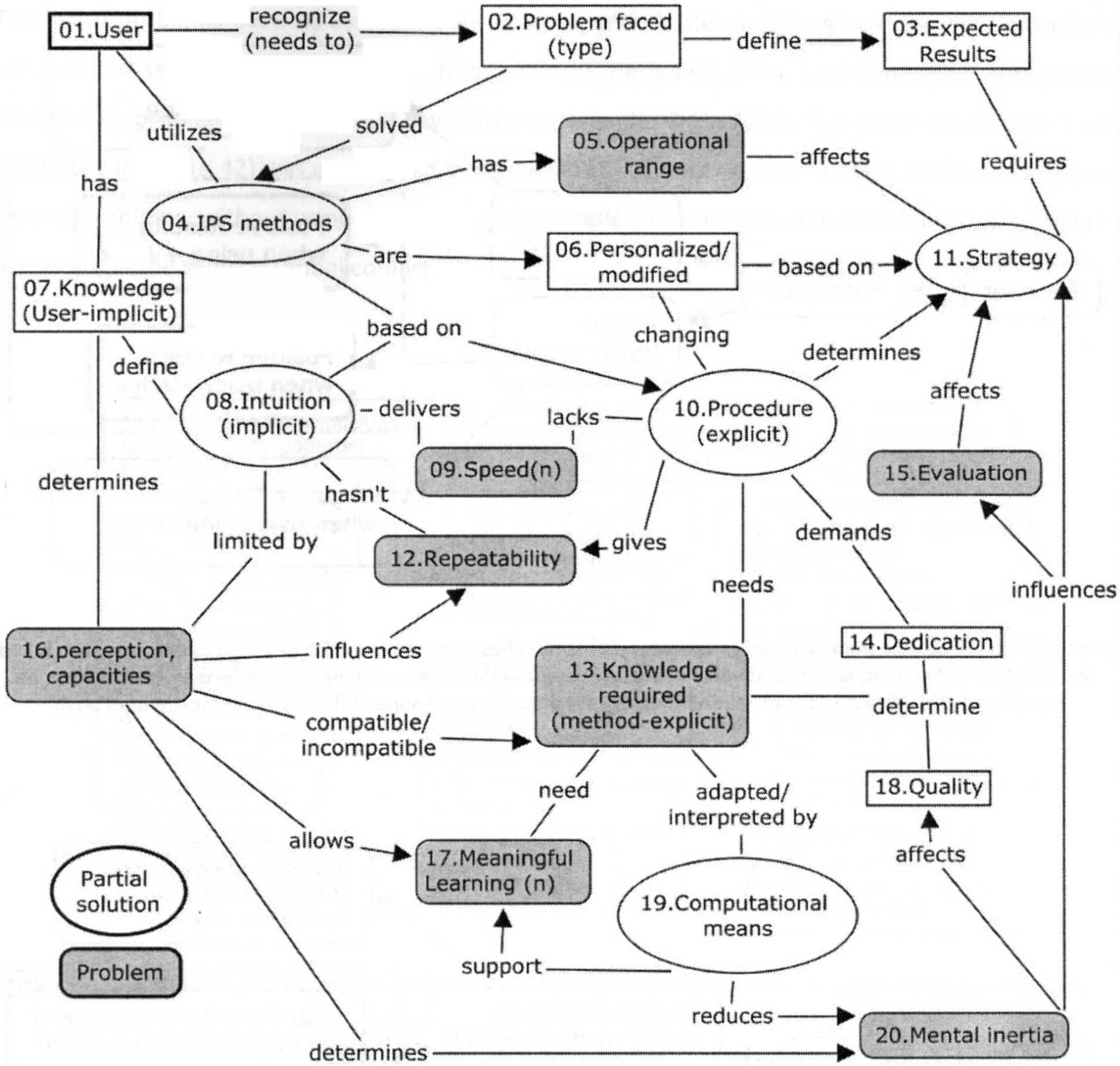

Figure 39: Network of problems, which includes partial solutions. The grey boxes indicates the problems, and the ovals indicate partial solutions.

3.2.2.3. Network of problems, expanded and with problems as contradictions

The problems presented in Figure 39 were reformulated as contradictions, utilizing the mini-problem format (Kucharavy 2006). An example of that reformulation is shown in Figure 41.

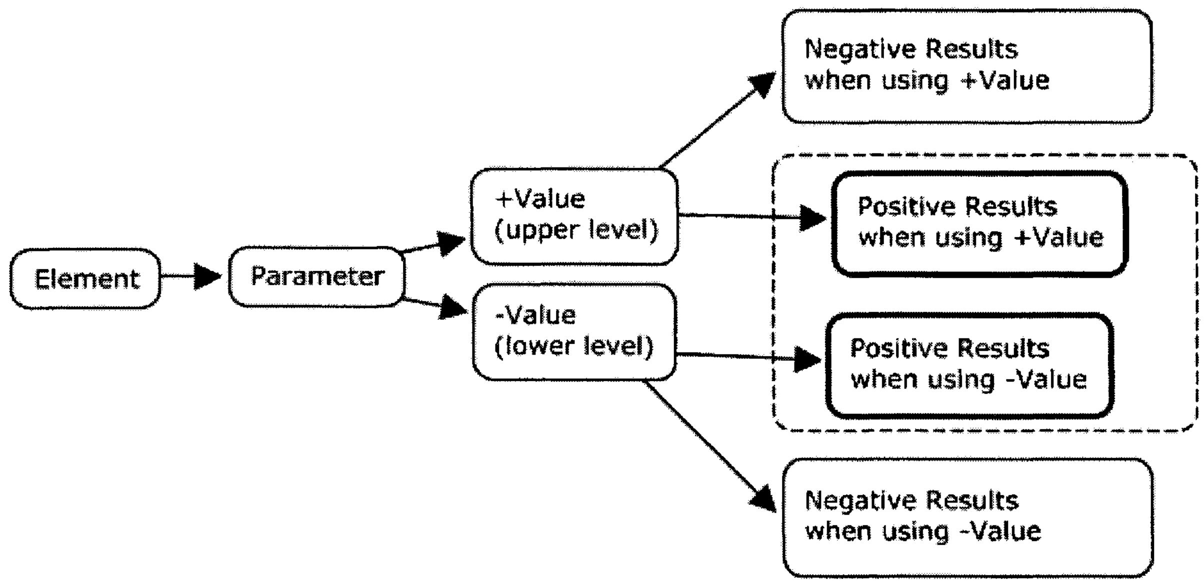

Figure 40: Contradiction diagram as a *mini-problem*: every element has parameters, which can have different values. In order to clarify the problematic, two distant values must be chosen, so the positive and negative effects of each one can be considered. The *Ideal result* is to have all the possitive outcomes and none of the negatives (segmented box).

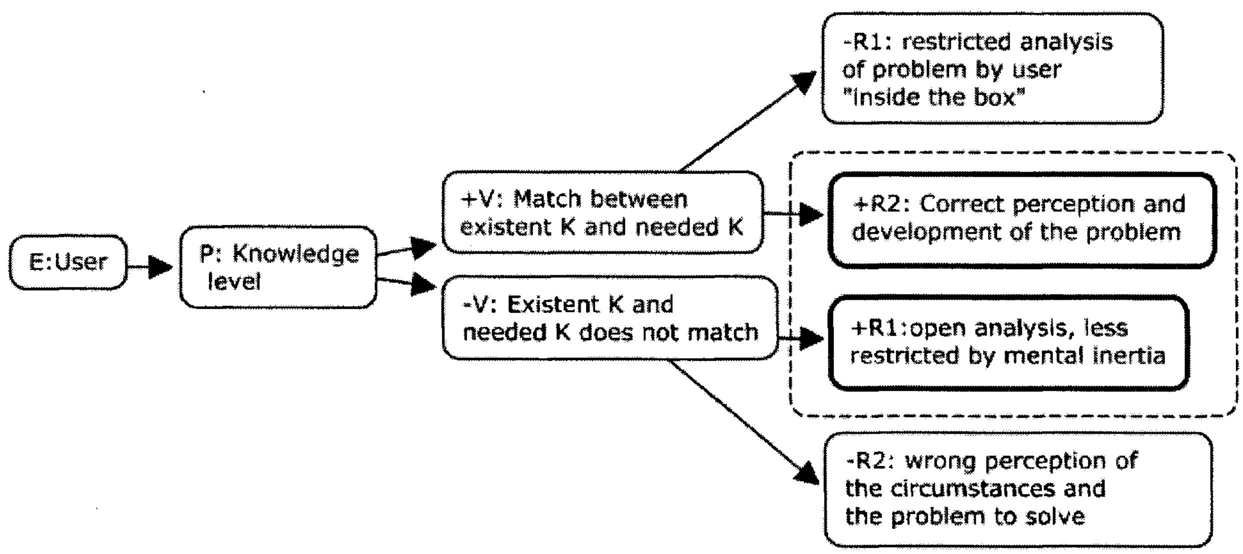

Figure 41: Each component of the NoP was analyzed as a mini-problem. In this example, problem 16 "perception, capacities" is related to the element *user*, specifically with its parameter *knowledge level* (k) which can match or not the needed one. The ideal result includes the correct perception and analysis of problems without mental inertia.

The list of problems (now as contradictions) was expanded after analyzing how the different methods presented in section 2.1 allowed confronting a specific design situation: how to design anthropometrics workplaces, problem presented and developed in (Helander & Lin 2000). The new expanded list (sample presented in Figure 42) was also turned into contradictions, process that resulted in the convergence of several problems, without using any sort of generalization. This indicated that the observed problematic has as origin the same essential conflict.

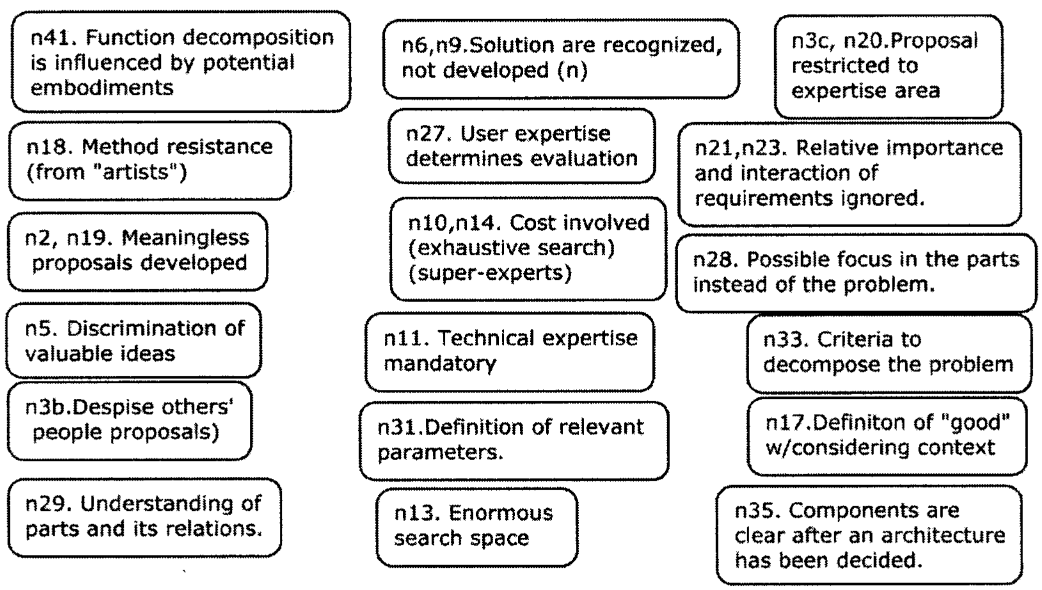

Figure 42: Partial list of problems regarding IPS methods, developed during the literature review presented in section 2.1.

In parallel, several relations between problems began to become evident. The development of those relations, necessary to understand the problems itself, resulted in the diagram presented in the third NoP (Figure 43). On it, many of the *problems* showed to be more *neutral,* with positive and negative effects.

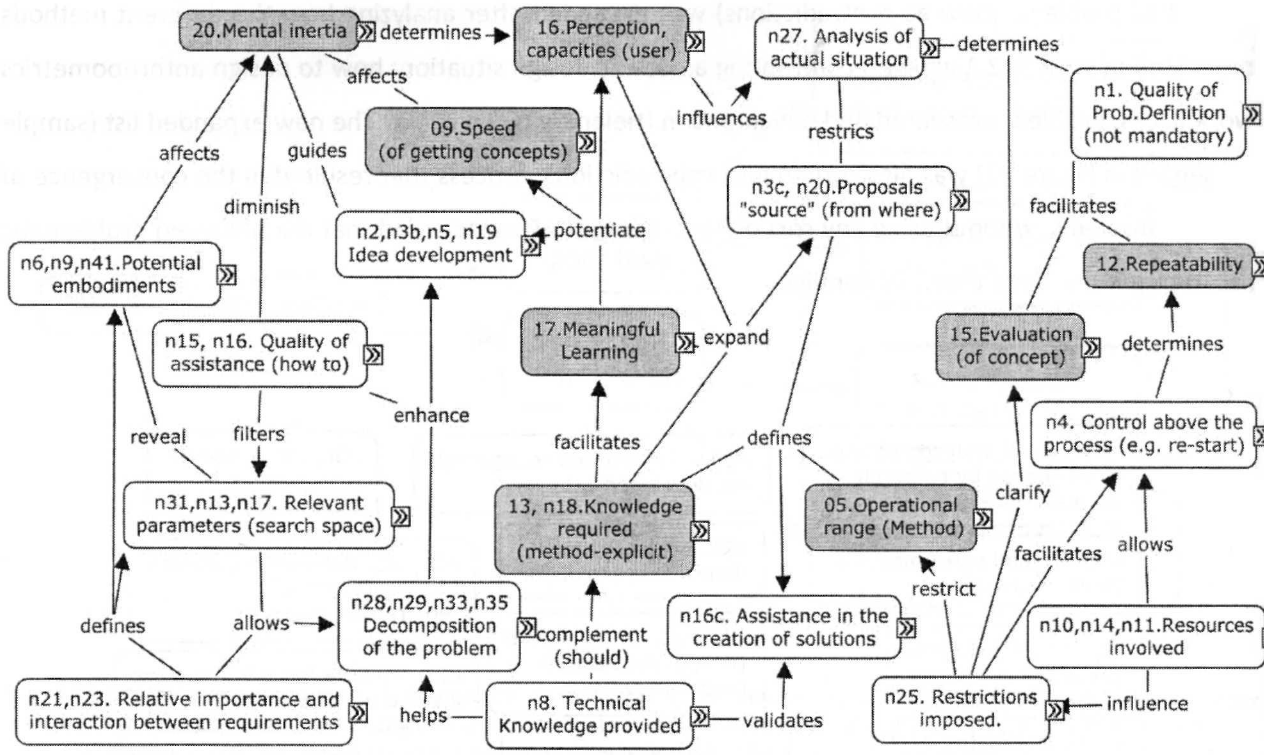

Figure 43: Third NoP, with problems as contradictions. The grey boxes are the problems obtained from the second NoP (Figure 39). The Details of Each contradiction can be verified in the attached DVD.

3.2.3. Revealed research questions

The activities described in sections and 3.2.1 to 3.2.2 clarifies the objectives and stages needed to understand IPS methods.

3.2.3.1. Question 1: Which are the requirements that an IPS method should fulfill?

To identify all the relevant characteristics of IPS methods is unlikely, but the critical ones that define its usefulness should be clear. This can be seen as the *problem statement stage* of IPS methods: if we don't know the requirements and the reasons behind the difficulties to achieve them, then no amount of work will be productive. For example, if inspiration were abundant and reliable no method would be needed in the first place. The important questions are the "Why?" behind the list of steps mentioned in literature in order to solve IP. Those are the requirements, consequence of the conflicts that need to be overcome.

This declaration of what has to be overcome will focus the research and allows to see how CS is being a support, which existing CS technology can be turned into support, and what needs to be developed in order to integrally assist IPS using CS.

3.2.3.2. Question 2: How today's methods are dealing with those requirements?

If all the requirements of question 1 answer are fulfilled by some method, then two alternatives are possible:

- There is no problem to be solved but a task to be performed.
- The list of requirements is missing an important component.

If a method does not deal with the most basic requirements of IPS, its effect is no better than a placebo that gives the illusion of doing something useful, reason why the actual method's strategies need to be matched against the requirements, paying special attention on how IPS dependence of specific individuals is handled (if considered at all).

3.2.3.3. Question 3: Which are the conflicts that impede to develop the missing requirements?

The most popular methods are almost effortless, but they offer no advantage to solve IP when compared to random inspiration ("no free lunch?"). If this happens is because there are conflicts behind the demanded requirements that has not been considered and need to be developed in order to support IPS.

3.2.3.4. Question 4: How Computer Science can achieve (contribute to) what actual methods can and can't do?

In concept development, the aid provided by CS is significantly less that, for example, design planning and embodiment. Since the requirements of IPS need to be fulfilled either by a local solution by a more radical change at a superior level of the system, CS could develop specific tools to level weak (or non-existing) areas, or to provide a new integral way of dealing with IP situations.

In general it can be said that there are several areas that can be investigated, but their relative significance and probably the order in the IPS process implies a natural sequence of development, i.e. without listing the requirements, any type of solution search will be random, reason why that is the next topic.

3.3.Requirements of a method for IPS

The activities performed (section 3.2) clarified the most relevant requirements (and conflicts behind them) of IPS methods, allowing studying how they are treated. This opened the path to CS potential contribution.

3.3.1. Most fundamental problems of IPS methods

The following are the (founded) fundamental problems that an IPS method has to deal with. The selection was made based how to control the whole NoP with its minimum parts.

For each one, an explanation and justification of its election is presented, followed by its correspondent mini-problem.

3.3.1.1. Analysis of the actual situation

The most common declared start point of IPS methods. If it's made with rigor is useful, reusable, and results are expectable, because it gives direction to any effort from the beginning. Without understanding the actual situation, there is no process but random search.

It is elected because is the first check point of a problematic, and defines and direct the need of generate (or not) concepts.

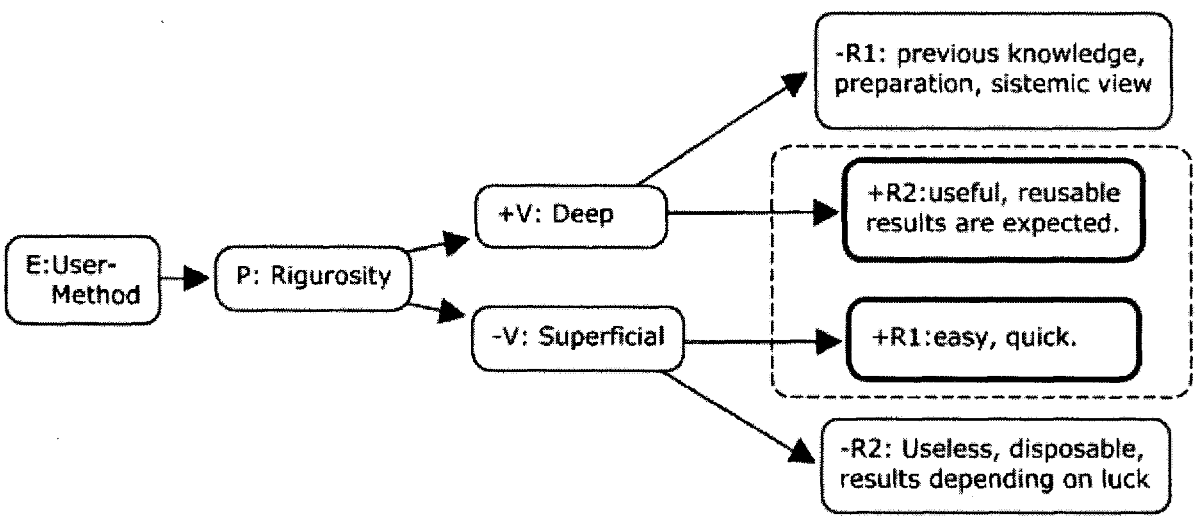

Figure 44: Mini-Problem of the *analysis of actual situation*. Ideally, it should guide to results, be reusable, quick, and easy to perform.

3.3.1.2. *Mental inertia*

A characteristic of the user is the natural an unconscious tendency to be carried by past experiences believing that nothing else is viable or needed to be known. It is desirable to be low in order to permit unfamiliar alternatives to become solutions, but to avoid it requires preparation and constant awareness. A method who doesn't treat it seriously will probably permit a quick implementation, but it usefulness will be (at most) a checklist of IPS activities.

It is elected because it concentrates a lot of causes, being a mayor influence to the user. To block it is essential to really explore the search space.

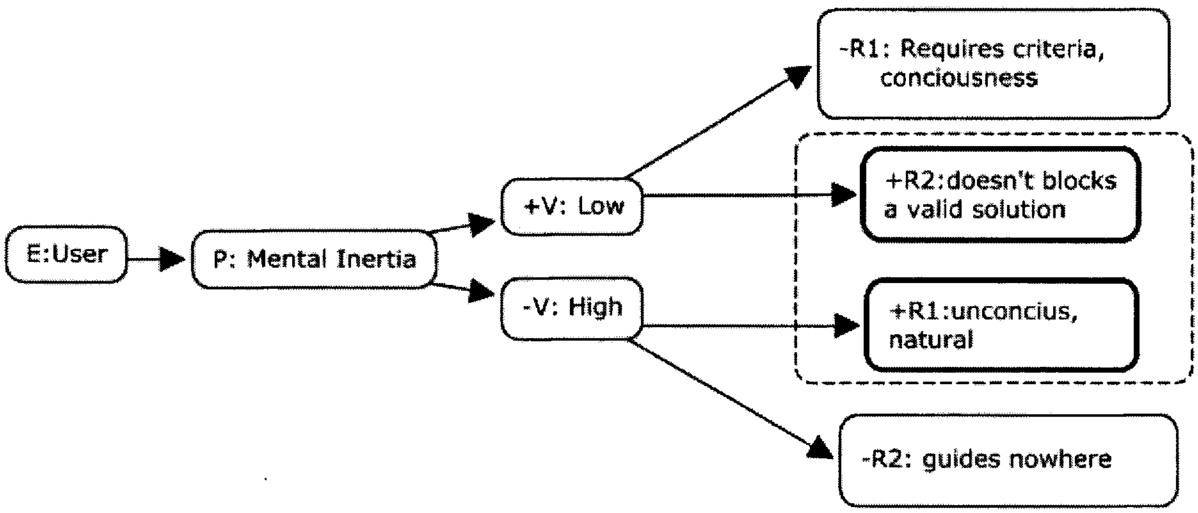

Figure 45: Mini-Problem of *mental inertia*. Ideally, the concept generation should be natural, but without blocking valid solutions caused by high levels of mental inertia.

3.3.1.3. *Meaningful Learning*

A characteristic of the kind of learning that the user developed. It is based on three components (Novak & Cañas 2008):

- Material conceptually clear
- Possession of prior relevant knowledge
- The decision of choosing to learn meaningfully

Without it, to utilize any method to understand the IP situation, visualize alternatives, and evaluate candidates, is (again) not a process but random search. As a consequence, a method that does not considers it as a need in its own learning process has a high chance of being used superficially or not being used at all.

It is elected because it defines the method usefulness, together with the user level of expertise (an expert has more meaningful learning than a novice).

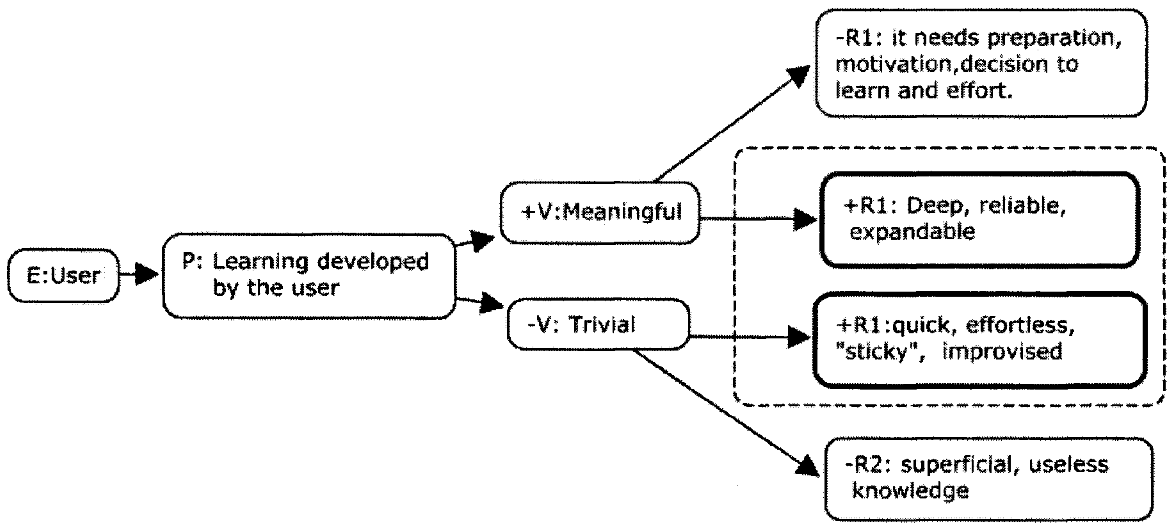

Figure 46: Mini-Problem of *meaningful learning*. Ideally, to learn a method should be quick and easy to keep, with a deep understanding of its aims and real capacities.

3.3.1.4. Assistance in the creation of solutions

Is the effective concept generation assistance that the method offers, the "how" it supports the creation of solutions. It should be reliable, traceable, and repeatable (the process), being this the real distinction of IPS methods.

It is elected because it justifies the method utilization.

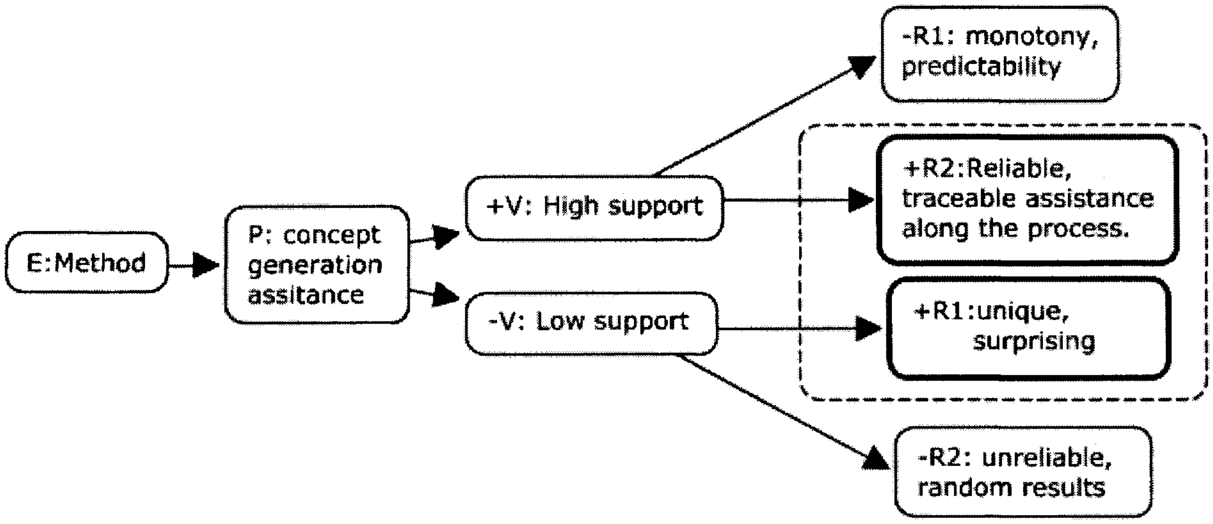

Figure 47: Mini-Problem of *assistance in the creation of solutions.* Ideally the method should give a reliable and traceable assistance that helps to deliver unique results.

3.3.1.5. Evaluation (validation) of the concept

Finally, an evaluation is necessary to determine and discriminate between valid concepts. If a method does not provide ways to evaluate the generated concepts, its usefulness will be conditioned to the particular ability of the user, depending on expertise.

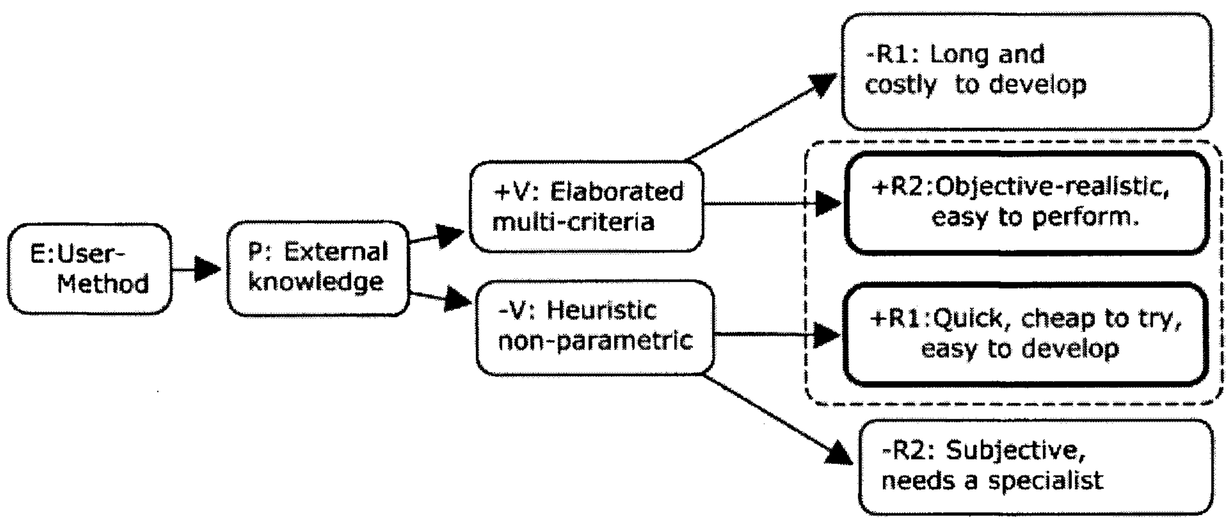

Figure 48: Mini-Problem of *evaluation (validation) of the concept.* Ideally, the method should give an objective, realistic, and easy to perform evaluation of candidates.

3.3.2. Actual methods strategies

When analyzing IPS methods, the central point is how they deal with the previously presented requirements. A method that does not treat them is considered, at most, a *tool* that can be helpful in specific situations. From the five requirements, *Meaningful learning* is the less treated in the IPS literature being considered a given most of the time or, when considered, treated superficially (e.g.: expert supervision needed, 50 hours of training needed).

Table 7 presents a checklist of the methods reviewed in section 2.1 versus the requirements considered as fundamentals, showing if they are considered or not.

Table 7: Central requirements in IPS v/s analyzed methods. This table shows if the method considers the particular requirement (+), considers it indirectly (~), or does not considers it at all (-). Quality or effectiveness are not evaluated.

	Analysis of the actual situation	Mental inertia	Meaningful Learning	Assistance in the creation of solutions	Evaluation of the concept
2.1.1 Heuristics	-	-	-	~	-
2.1.2 Brainstorming	-	-	-	~	-
2.1.3 Synectics	-	~	~	+	~
2.1.4 Conceptual decomposition	~	~	-	+	-
2.1.5 Morphological analysis	~	~	-	+	-
2.1.6 Axiomatic design	-	~	-	~	+
2.1.7 C-k Theory	+	~	~	-	~
2.1.8 Concept Generator	-	~	-	+	-
2.1.9 TRIZ	+	+	~	+	~
2.1.10 SIT	+	+	~	~	+

From looking the table several observations can be made, for example that heuristics and brainstorming do not deal directly with any of the requirements, reason why its performance is not different that random trial and error. On the other side of the spectrum it can be said that TRIZ is the most complete (not necessarily the best).

For each requirement the strategies proposed by the methods are:

- Analysis of the actual situation: In C-K theory, the concept-knowledge distinction clarifies and guides the needed development. SIT conditions force the same effect, and TRIZ provides specific tools to clarify the present state (like the *system operator*).
- Mental Inertia: SIT *close world* condition impedes to import pre-conceived solutions, encouraging the comprehension of the problem. TRIZ also deals with this issue using the concept of *contradiction*, which force the understanding of the causes that originates the problem.
- Meaningful Learning: from the studied methods, no one deals with this requirement directly.
- Assistance in the creation of solutions: Morphological Analysis and Conceptual decomposition combination search shows the potential alternative that, like Synectics analogies, can make the user to look into unexplored spaces. TRIZ matrix and standard solutions fulfill the same role.
- Evaluation of the concept: Axiomatic design independence and information axioms can be used to evaluate concepts. SIT conditions also gives a criteria to separate optimization proposals from creative ones.

Two mistakes (at least) can be induced by the table:

- To believe that applying two methods simultaneously (e.g. SIT and Ax. Design) all the requirements are being considered,
- To believe that most complete methods are going to deliver better results.

The analysis performed is a checklist, on which the quality and effectiveness of the listed methods are not evaluated.

3.3.3. Potential CS contribution

As seen in the *system operator diagram* (section 3.2.2), there are several stages where CS can improve its level of contribution, in order to balance the different components of the respective level to integrate them (as speculated in the 2031 window). Several aids are done regarding the principal requirements:

- The *analysis of the actual situation* requires a human to start it. Once a process has been modeled, the analysis can be done from many points of view (e.g. mechanical or economical), being possible to find potential areas of improvement.

- *Mental inertia* is not applicable in CS, but the search space is fixed. Consequently, the activity performed is optimization, thus in order to solve IP those borders should be variable by some criteria.
- *Meaningful learning* has been aided by CS, making the most lengthy and monotonous sub-tasks automatic, but since this is a user's characteristic, it cannot be developed by CS. Then the alternative is to develop computational means able to reduce the needed Meaningful learning to a minimum that can be achieved by any user.
- To *assist the creation of solutions*, CS gives real time support through databases and inspirational suggestions. There are also results in solving specific technical problems that are being perfected and hopefully expanded (like (J. R. Koza et al. 2008)).
- The *concept validation* has been aided by CS simulations, including the restrictions of each part of the system, requiring increasing the level of integration with the other stages of *concept development.*

In general, *mental inertia* and *meaningful learning* are the most open areas in IPS to be developed by CS. The most generic alternatives are to emulate the way that brain works or to find a totally different way to reach the same end (satisfy the requirement.

3.3.4. Relevance of the study of conflicts

The first benefit of this section is the clarification of the real *state of the art* regarding IPS: if an IPS method doesn't consider the critical aspects presented along section 3.3, then to study it is not a priority for this dissertation.

The original objective of using CS to assist *concept generation* has not change, but a new reason emerges: there must exist a balance between method and user, that finally determines the usefulness of the method, and this balance is dictated by the **conceptual generating capacity of the user and the method together, as a system**, and an automated system is the ideal, because can be utilized by all kind of users. This idea is developed further in section 3.4.1.

3.4.IPS derived ideas

In order to solve IP, it is necessary to learn and develop what is not provided by the method. For example, *Brainstorming* requires learn the whole system interactions in order to develop valid proposals, and *Morphological Analysis* needs the knowledge about valid solutions in order to compare.

- *Mental inertia* is not applicable in CS, but the search space is fixed. Consequently, the activity performed is optimization, thus in order to solve IP those borders should be variable by some criteria.
- *Meaningful learning* has been aided by CS, making the most lengthy and monotonous sub-tasks automatic, but since this is a user's characteristic, it cannot be developed by CS. Then the alternative is to develop computational means able to reduce the needed Meaningful learning to a minimum that can be achieved by any user.
- To *assist the creation of solutions*, CS gives real time support through databases and inspirational suggestions. There are also results in solving specific technical problems that are being perfected and hopefully expanded (like (J. R. Koza et al. 2008)).
- The *concept validation* has been aided by CS simulations, including the restrictions of each part of the system, requiring increasing the level of integration with the other stages of *concept development*.

In general, *mental inertia* and *meaningful learning* are the most open areas in IPS to be developed by CS. The most generic alternatives are to emulate the way that brain works or to find a totally different way to reach the same end (satisfy the requirement.

3.3.4. Relevance of the study of conflicts

The first benefit of this section is the clarification of the real *state of the art* regarding IPS: if an IPS method doesn't consider the critical aspects presented along section 3.3, then to study it is not a priority for this dissertation.

The original objective of using CS to assist *concept generation* has not change, but a new reason emerges: there must exist a balance between method and user, that finally determines the usefulness of the method, and this balance is dictated by the **conceptual generating capacity of the user and the method together, as a system**, and an automated system is the ideal, because can be utilized by all kind of users. This idea is developed further in section 3.4.1.

3.4.IPS derived ideas

In order to solve IP, it is necessary to learn and develop what is not provided by the method. For example, *Brainstorming* requires learn the whole system interactions in order to develop valid proposals, and *Morphological Analysis* needs the knowledge about valid solutions in order to compare.

This *balance* between user and method finally determines the usefulness of the method, and is not being considered in IPS methods. Looking from the previous perspective, the assisted generation of concepts needs to be able to be balanced according to user, being the most general method the most automated one. Two fundamental ideas converge into a proposal (section 3.5) to deal with the need of a more automatic *concept generation.*

3.4.1. Meaningful learning balance

Meaningful learning seems to be fundamental to first utilize and then to adapt (improve) a method, but the preparation and work that this decision implies makes most of the users to return to more generic-intuitive methods.

It exist a contradiction between having a deep & useful knowledge and to obtain it quickly & effortless. The proposed idea is to explore the conditions to solve this contradiction by computational means, diminishing the human involvement aspiring that the problem is solved with no effort[9]. But, despite that the computational IPS is a desirable objective, it is not a mandatory neither necessary one.

What it seems to be necessary, and thus a potential research direction, is **the *balance* between the user level of expertise and the degree of detail of the method**. For example, a *visionary* is unlikely to need a method to analyze and solve a situation of his field; however, a *novice* needs a step by step procedure in order to advance.

On Figure 49, there are presented the main ideas of balance. On the diagram, if one imagines a horizontal line it can be obtained a qualitative perspective of the kind of balance between user and method. For example in the base, the method importance is almost total, needing explicit knowledge in order to deal effectively with problems. The situation at the top is the opposite, where the individual does not needs a method and can work effectively with abstract ideas and heuristics. It also shows that the need of an expert can be supplied by automatic methods if the conceptual ideas are turned into explicit knowledge, and that the method importance diminishes if the novice evolves into an expert.

[9] This is coherent with TRIZ laws of evolution: less human involvement, increasing system's *ideality*.

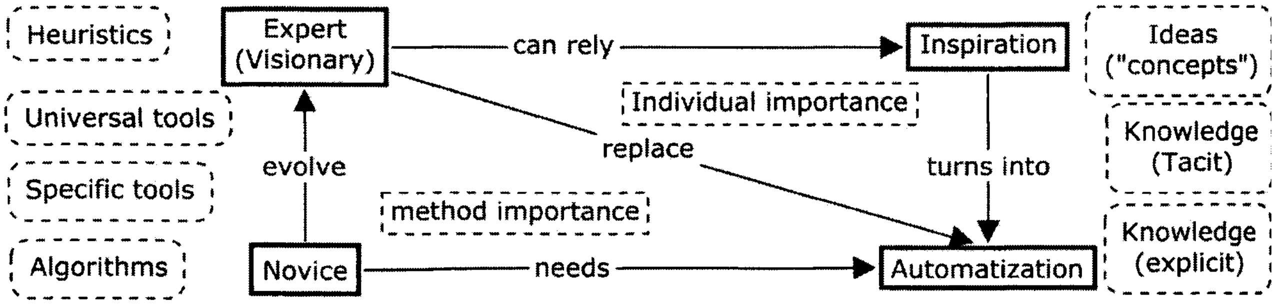

Figure 49: *Balance* main ideas. In the base, the method importance is almost total, in oposition to the top, where the individual importance is absolute.

From this point of view, it can be argued that the dependence of expertise and intuition (a high level of *meaningful learning*) can be managed increasing the detail level of the method, which extended to an extreme means to eliminate the need of criteria, allowing the automation of IPS. To move the balance line to the bottom implies to understand how to develop concepts without human involvement.

Without renouncing to the previous, to study the characteristics that a method requires in order to be useful moving along the scale of expertise, seems interesting and useful to understand IPS evolution.

3.4.2. Evolutionary algorithms development

In order to solve problems without human involvement, concepts must be developed by computers. On this area EAs have shown promising results (like in (Bentley & Wakefield 1997a; J. Koza et al. 2004; Williams 2005)).

As explained in section 2.2, EAs follow the idea of biological evolution within a computational framework, being two of their most popular examples the Genetic algorithms and Genetic Programming. Both methods require the definition of the input (parameters) and output (objective), but **GAs requires additionally the codification of the parameters** (the model). This last characteristic defines the relation between parameters, situation inexistent in GP, who search precisely the best possible relation. If a variable is not relevant, GP ignores it during its evolution, allowing adding a few parameters in case of doubts about its relevance (Poli et al. 2008). By doing this there is a possibility to bias the system, but that can be said about every action in IPS situations.

Considering the previous, a potential research direction is the development of qualitative changes within EAs, in order to generate concepts. First, the requirements before modeling, second a model proposal, and third an implementation to be studied.

One important observation is that still today is not completely clear why EAs generate high-fitness solutions when applied to practical problems. The dominant theory is the building block hypothesis (BBH), which basically states that the codification allows identifying and recombining "building blocks", i.e. short parts of the code with above average fitness, which reduce the complexity of the problem creating optimal sub-solutions that later are recombined to deliver good general proposals (David E. Goldberg 1989). However, there are several authors claiming that BBH is neither convincing nor rigorous (O'Reilly & Oppacher 1995), and many others that continue to develop the concept (McPhee et al. 2008).

3.5.Model proposal

After the study of the IPS problems, together with the literature review, it is decided to utilize *dialectics* as the basis to develop the fundamentals of a computer assisted method able to aid the conceptual development of products. In essence, it can be said that *dialectics* is the philosophy of how the transformations operate, and in this dissertation context, a dialectical framework can give to a problem solving methodology (in this case TRIZ) the necessary structure to be developed further. Moreover the dialectical philosophy, materialism in particular, is completely consequent with the spirit of automate the concept generation[10] since in a materialistic mind there is no such thing as "divine inspiration". This shouldn´t be understood as a desire to eliminate the spontaneous creativity.

The macro strategy can be resumed in:

- Dialectical principles are the conceptual framework to understand and cross the quantitative-qualitative barrier in IPS.
- EAs are the mechanism to implement the previous computationally.
- TRIZ concepts are the means to direct the search, in order to make computationally viable the process of modify the search space and its restrictions.

The following sub sections clarify and justify the terms and ideas mentioned.

[10] TRIZ objective of controlling the process of creativity points toward the same objective (Altshuller 1984).

One important observation is that still today is not completely clear why EAs generate high-fitness solutions when applied to practical problems. The dominant theory is the building block hypothesis (BBH), which basically states that the codification allows identifying and recombining "building blocks", i.e. short parts of the code with above average fitness, which reduce the complexity of the problem creating optimal sub-solutions that later are recombined to deliver good general proposals (David E. Goldberg 1989). However, there are several authors claiming that BBH is neither convincing nor rigorous (O'Reilly & Oppacher 1995), and many others that continue to develop the concept (McPhee et al. 2008).

3.5.Model proposal

After the study of the IPS problems, together with the literature review, it is decided to utilize *dialectics* as the basis to develop the fundamentals of a computer assisted method able to aid the conceptual development of products. In essence, it can be said that *dialectics* is the philosophy of how the transformations operate, and in this dissertation context, a dialectical framework can give to a problem solving methodology (in this case TRIZ) the necessary structure to be developed further. Moreover the dialectical philosophy, materialism in particular, is completely consequent with the spirit of automate the concept generation[10] since in a materialistic mind there is no such thing as "divine inspiration". This shouldn´t be understood as a desire to eliminate the spontaneous creativity.

The macro strategy can be resumed in:

- Dialectical principles are the conceptual framework to understand and cross the quantitative-qualitative barrier in IPS.
- EAs are the mechanism to implement the previous computationally.
- TRIZ concepts are the means to direct the search, in order to make computationally viable the process of modify the search space and its restrictions.

The following sub sections clarify and justify the terms and ideas mentioned.

[10] TRIZ objective of controlling the process of creativity points toward the same objective (Altshuller 1984).

3.5.1. Essentials of Dialectics

The modern concept of dialectics goes back to Hegel. His main contribution is the proposal of "dialectic evolution", which contains the idea of the inherent contradiction inside every "entity". This contradiction is the source of the entity self-movement and thus its evolution (Lewis et al. 1932).

Hegel´s system can be criticized by several aspects, but the more important in this dissertation context is the belief that "ideas precede matter" and that the search for perfection is mental, which implies that perfection already exists, and all the changes are just appearance.

This idealistic approach is later reconstructed by Marx. First he admits that the world is far from perfection and the changes are real, from confusion towards order, irrational towards rational. The second main modification is the idea of "materialism", that it must be understood as the recognition to the fact that matter exist, and a matter who think (ideas) it has its origin in a matter who doesn´t. Is a concept to designate the objective reality, exterior to our consciousness and independent of it. The mental and spiritual manifestations (art, feelings, others) doesn´t come from another "hidden dimension", they are the result of eons of non-thinking matter existence (Lewis et al. 1932) . An inevitable conclusion is that the **matter has the faculty of transform itself, to "evolve"**.

Based on the previous, Marxism sets out to answer the most basic questions related to both, nature and humanity: the origin of life, species, evolution, history, consciousness and mind .The basis to respond those questions are the three laws of motion, which are part and can be discovered within the entire universe.

3.5.1.1. Three laws of Dialectics

The order and name of how dialectic laws are presented differ among sources, but without substantial changes in their meaning. They are as follows (F. Engels et al. 1964):

a) The law of the transformation of quantity into quality and vice versa.

Objects and phenomena are characterized by qualitative and quantitative aspects. Qualitative aspects refer to the distinctive qualities that define an entity conceptually, while quantitative aspects are related to magnitudes, and aren´t related to the essence of an entity. The law or transformation states that continuous quantitative development results in qualitative changes sooner or later, happening in "leaps" that interrupt the normal rhythm of development, being its cause the long process of accumulated quantitative changes (Yajot 1973).

As an example, if a new kind of wheat where going to be developed, there will be a lot of experiments, every one slightly different from the others, then the most productive one (quality) will be selected to increase the amount of wheat (quantity).

b) The law of the interpenetration of opposites.

Also known as the "law of unity and struggle between opposites", it basically states that everything in nature is composed of pairs of opposites that together create an indissoluble unity (e.g. light and darkness). This kind of though can be found in millenary books[11], and there are plenty of understandable examples: electricity positive and negative charge, light and darkness, birth and death, and so on. This fight can be understood as the aspiration of each one of the opposites to be the dominant inside this unity. If we look at galaxy scale, a star is held together by gravity trying to pull all the molecules to the center, and heat trying to send them as far from the center as possible. If either force is completely successful the star ceases to be, if heat wins a supernova is created, if gravity is victorious a black hole shows up. At human scale, living things balance internal and external opposing forces (acidity and alkalinity) to maintain homeostasis.

The dialectic conclusion is that these two opposites conform an unique contradictory process, and their existence is reciprocal. This can be also understood in other sense, the "identity": under certain conditions, the opposites turn into each other (Yajot 1973). As an example, the warm object gets cold and vice versa. Behind this unity and contradiction lies the reason that makes each entity dynamic providing constant motivation for movement: the struggle between opposites.

c) The law of the negation of the negation: Everything in nature after coming into being grows, becomes mature, and then dies. The law of negation accounts for this tendency that leads to increasing the quantity of all things as a result. This is complementary to the law of opposites, which produces conflicts in each element and gives it motion, to later negate its own nature. This dynamic process of birth and destruction is what causes entities to advance, to evolve. This is also referred to as the cycle of thesis-antithesis-synthesis, in a spiral path from the inferior towards the superior.

3.5.2. Proposed model of IPS

It has been mentioned before that this dissertation is focused and the *conceptual generation* stage of product development. Figure 50 offers a concept map (Novak & Cañas 2008) of the IP solving process under the dialectical negation perspective, with the critical tasks to be performed.

[11] Probably the oldest one is the "Tao te King", attributed to Lao-Tse, which is the base of Taoism.

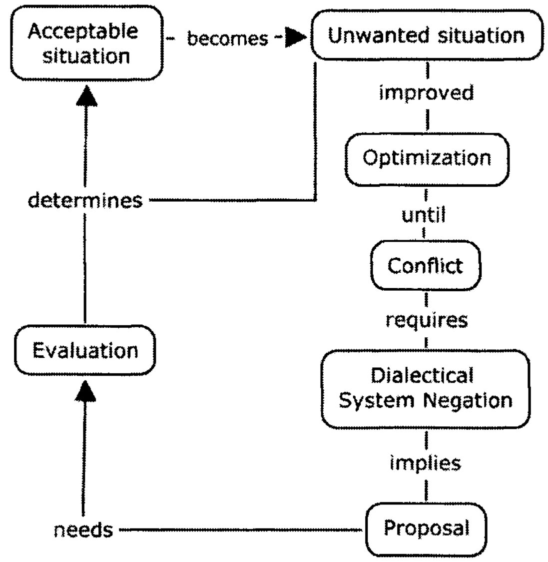

Figure 50: IP solving generic stages

The descriptions of the generic stages in the IP solving cycle are the following:

- Unwanted situation: situation to be changed, with at least one obstacle to doing it. It starts the IPS process.
- Optimization: improvement of the unwanted situation by quantitative changes, meaning without modifying the system (elements or their relationships), just the magnitudes of the parameters involved.
- Conflict: when quantitative changes no longer deliver satisfactory performances, and to improve further one parameter negatively affects the performance of at least one other.
- Dialectical system negation: significant modification to the current system, in order to deliver a satisfactory performance.
- Proposal: emergence and selection of a new valid concept generated from the negation of the system.
- Evaluation: Evaluation of the selected proposal, to identify if the system reaches a new desired state or a new unacceptable one. If the evaluation is satisfactory, the cycle stops until a new unwanted situation emerges.

Although in previous optimization approaches some of these generic stages are also implemented, the concepts of *conflict* as a restriction of the search space for IP solution and *system negation* to generate new conceptual proposals are, together, a new characteristic of the *dialectical negation algorithm*[12] (DNA) approach. Other optimization-based approaches suggest simply widening the search space when no satisfactory solutions are found in the space given initially. However, widening is neither simple nor

[12] Rigorously, the DNA is a pseudo-algorithm since it lists the activities to perform, not calculating any function.

effective for IP solving because this way the search space, and therefore the search time involved, may become infinite. It also implies understanding the reasons behind the constraint to be edited/deleted, being the most probable explanation the existence of a background conflict (manufacture, legal, aesthetic). If expanding the search space demands the solution of another IP, it cannot be considered a good alternative, or at least not a trivial one.

To explain in more detail the activities between the emergence of a *conflict* and the generation of a *proposal*, a more detailed IP solving conceptual map is shown in Figure 51.

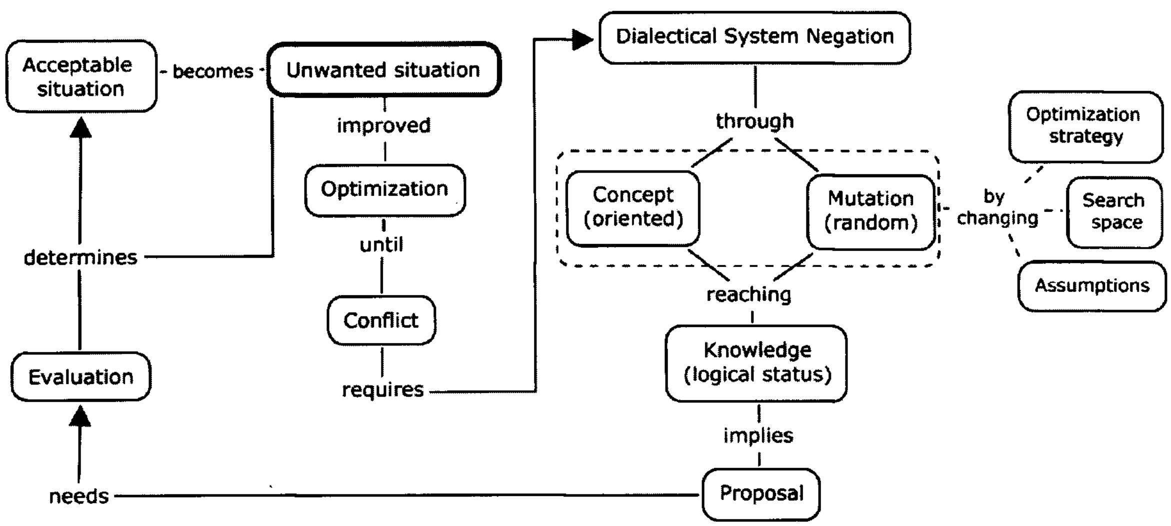

Figure 51: The expansion between the negation and the proposal is where the computer support is intended.

The dialectical system negation can be achieved by a standard mutation (random, like blind trial & error) or guided by a concept (using a predefined method). In both situations the system negation necessarily implies to modify at least one of the following:

- Optimization strategy: to change the way that the search for alternatives is done. For example, instead of using design of experiments (Antony 2003), utilize GAs in order to find the best combination of values (See Figure 52).
- Search space: to modify the range or quantity of variables. For example, instead of searching 2D shapes using depth as a constant, to include depth as a variable will deliver a new dimension to evaluate candidate solutions (See Figure 53).
- Assumptions: to modify one of the initial conditions of the problem, thus changing the problem to be solved. For example, allowing the use of materials that were considered forbidden originally (See Figure 54).

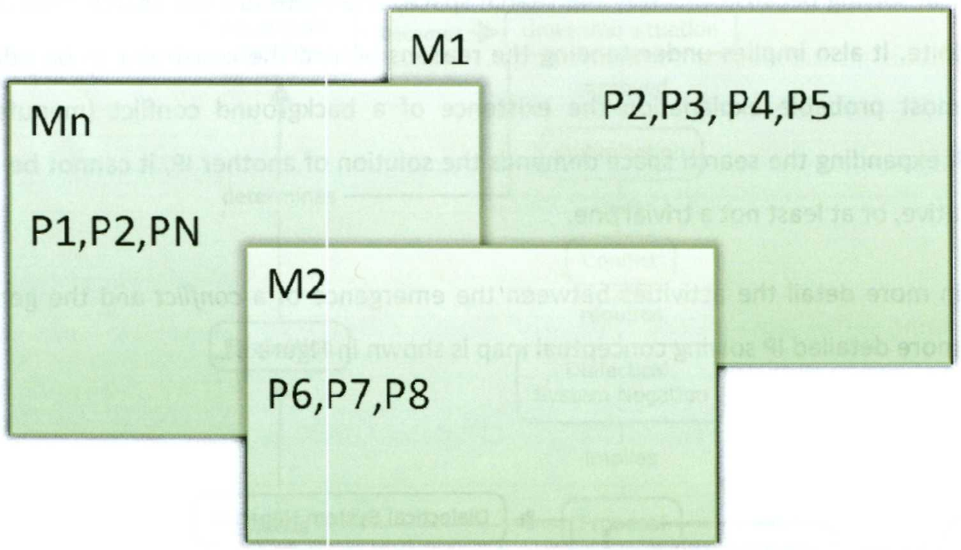

Figure 52: Negation of the system by changing the search (optimization) method. Each method will deliver its own lists of parameters values restricted to the ranges defined.

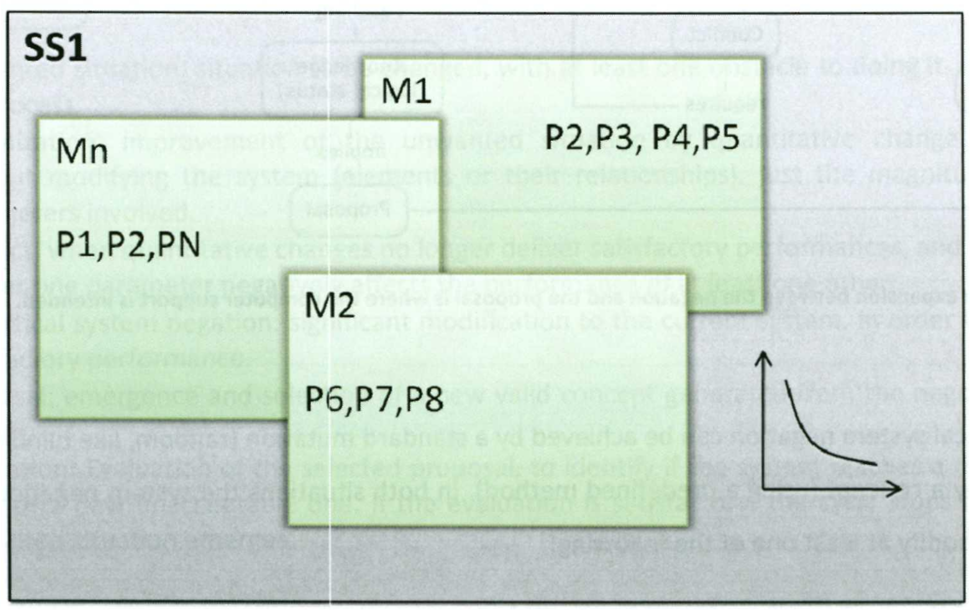

Figure 53: Negation of the system by changing the search space, modifying the parameters considered. The inclusion of a new parameter implies a new system Pareto front that can fulfill the intended conditions.

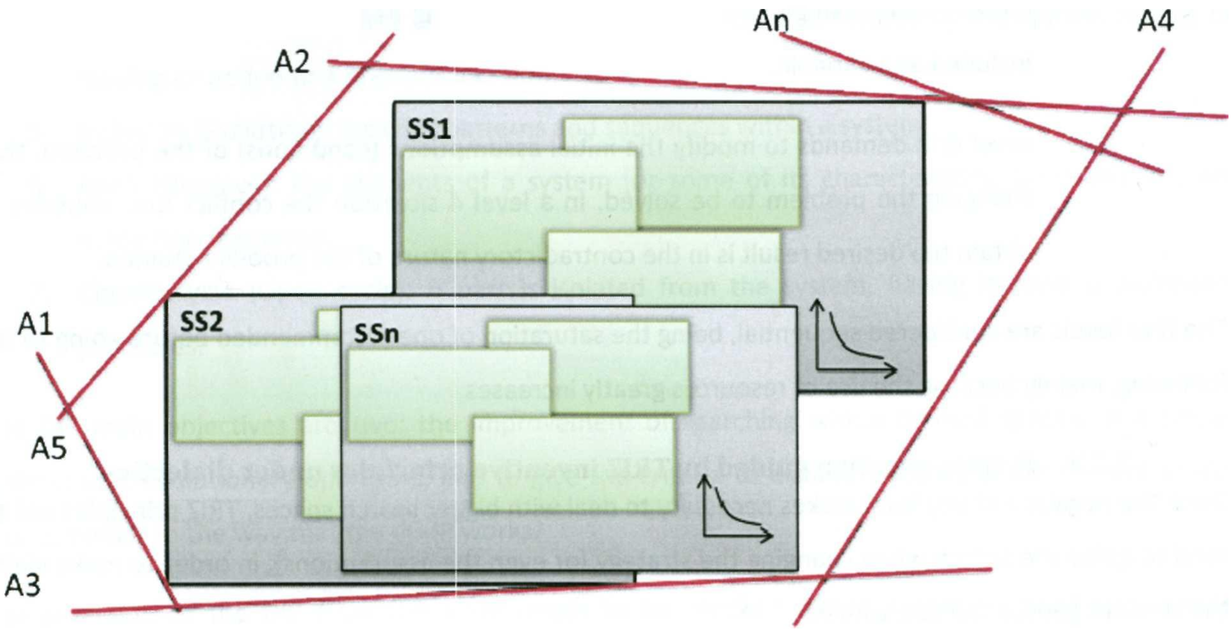

Figure 54: Negation of the system by changing the assumptions, modifying the search space and also the problem in particular.

Based on the previous classification, the required level of inventiveness (or problem level) can be defined, from an analytical point of view, observing the kind of needed modification:

- Level 1: The solution of the problem does not require any negation of the system components, being possible to obtain it by adjusting the parameters using a known optimization procedure or standard technique. This can be performed until the situation Pareto front is reached.
- Level 2: it demands to change the way that the search for alternatives is performed within the system. In a level 2 situation the analyzed system has reached its Pareto front without giving the desired performance. In order to explore differently the search space, the simplest option is to change the search **method**, mainly because this alternative doesn't have an influence in the performance of the problem variables.
- Level 3: it demands to modify (range or quantity) the considered **parameters**, meaning this the search space. Similarly to a level 2 situation the desired performance is not reached, but the different strategies does not deliver different results. In this situation the needed negation demands to define a criterion to expand (edit) the search space.

For example, instead of searching 2D shapes using depth as a constant, depth can be included as a variable.

- Level 4: it demands to modify the initial **assumptions** (conditions) of the problem, thus changing the problem to be solved. In a level 4 situation the conflict that impedes to obtain the desired result is in the contradictory nature of the problem studied.

The four levels are considered sequential, being the saturation of one recommended before going to the following, mainly because the use of resources greatly increases.

3.5.3. System negation guided by TRIZ inventive principles under dialectics

Since the negation at any level makes necessary to deal with bigger search spaces, TRIZ principles will be used to guide the search when changing the strategy (or even the assumptions), in order to make viable the concept generation as a whole.

Based on the 40-IP presented in section 2.1.9, a group of dialectical principles was developed using the following ideas ("axioms"):

- Inventive principles can be classified in a small number of generic categories, which can be made more specific as a result of "hybridizations" of higher level categories (Negation).
- Advances related to generic levels will influence the more specific ones, and vice-versa (Transformation).
- Every category is constituted by a pair of opposites (Opposites).

The selection of the 40-IP as a start point is based on its empirical origin (patent analysis) and its popularity between TRIZ practitioners. They were arranged through an affinity diagram (Otto & Wood 2001), aiming to first minimize the overlap between principles, and second to construct a more intuitive and practical format, reducing in practice the amount of information handled by increasing the logic of the tool. This result in the following 7 Dialectical Principles (DiP):

1. **Separation** (Unification): The different elements of a system are made more independent related to each other.
2. **Minimize range** (Maximize): Affects the extension of the actions applied to a system (partial, general, concentrated, diluted, etc).
3. **Balance** (Unbalance): it distributes uniformly the components or actions within a system.

4. **Pre-action** (Post-action): it affects the sequence of events performed by the system, adding or moving an action to a previous location.
5. **Inclusion** (Exclusion): includes patterns and sequences within a system.
6. **Align** (Displace): the elements of a system (or some of its characteristics, like materials) are made homogeneous.
7. **Closed cycle** (Open cycle): A part is isolated from the system, having its own evolutionary processes.

The DiP main objectives are two: the improvement of searching within defined spaces, in a similar fashion that evolutionary operators, and to give the criteria to expand (negate) an insufficient search space, similar to the way that the 40-IP works.

The reversion of the DiP from the 40-IP needs to be checked for consistency against the *laws of dialectics*, since *they* are the framework intended to give coherence to the hybrid development of IPS methods and EAs.

1. **The law of the transformation of quantity into quality and vice versa:** The dialectical model (Figure 51) is based in that the first stage in IPS is to reach a conflict (Pareto front), where quantitative improvements are no longer useful, indicating that a qualitative change is necessary. The law of transformation clearly shows that the system must be negated, and the DiP are the developed tool to perform the change of strategy, space, or assumptions.

2. **The law of the interpenetration of opposites:** All DiP are dual, being the tools to negate (evolve) a system. This is something that is not present in the 40-IP despite the dialectical influence on TRIZ origin and development (Dung 1997). Also, the whole model is a closed loop where the separation and union of the opposites (physically or temporarily) is considered all the time.

3. **The law of the negation of the negation:** The evolution of a system negating itself (its strategy, space, or assumptions) is especially clear in the DiP. For example, if a system component is not working as intended and one DiP is identified to be in one extreme, it is expected to explore the other side; e.g. if a "maximum range" action has negative consequences it can be "minimized", a "balanced" component can be "unbalanced", and so on.

Originally, the DiP were thought to be utilized together with 2 complementary operators: Physical/Temporal, and Integrated/Disintegrated (Duran-Novoa et al. 2009). However later this was discarded for being part of the second and third dialectic laws: **the cyclic nature of everything,** which separates and unifies in both time and space. Additionally, the separation Physical/Temporal is arbitrary since reality is constituted of 4 dimensions, and they should be considered always simultaneously.

All the 40-IP were grouped under an equivalent DiP, being necessary in many cases to list a complementary one, due to the different range of detail found in the 40-IP. Two principles, "blessing in disguise (22)" and "self-service (25)", are not included in this categorization since they are considered declarations of the ideal final result and consequently they should be considered in every situation (as a general advice).

The result of this grouping is presented in Table 9 and Table 8, the first one being more specific regarding the principles involved.

Table 8: Correspondence between the 40-IP and the DiP.

Separation / Unification	Minimize/ Maximize range	Balance / Unbalance	Pre/ post-action	Inclusion/ exclusion	Displace / Align	Closed / Open Cycle
1	3	4	9	28	12	13
2	6	8	10	29	14	23
5	16	31	11	38	17	26
7	20	33	32	39	30	27
15	36	40			35	34
18	37					
19						
21						
24						

Table 9: Equivalence between TRIZ 40 Inventive principles and the seven dialectical ones.

40-IP	Equivalent DiP	Complementary DiP
Principle 1. Segmentation	Separation	
Principle 2. Taking out	Separation	Closed cycle
Principle 5. Merging	Unification	
Principle 7. "Nested doll"	Unification	Inclusion
Principle 15. Dynamics	Separation	Close cycle
Principle 18. Mechanical vibration	Separation	Close cycle
Principle 19. Periodic action	Separation	Close cycle
Principle 21. Skipping	Separation	Open cycle
Principle 24. 'Intermediary'	Separation	Inclusion
Principle 3. Local quality	Min. Range	Closed cycle
Principle 6. Universality	Max. Range	
Principle 16. Partial or excessive actions	Max. Range	Post action
Principle 20. Continuity of useful action	Max. Range	
Principle 36. Phase transitions	Max. Range	
Principle 37. Thermal expansion	Max. Range	
Principle 4. Asymmetry	Unbalance	
Principle 8. Anti-weight	Unbalance	Pre Action
Principle 31. Porous materials	Unbalance	Align
Principle 33. Homogeneity	Balance	
Principle 40. Composite materials	Unbalance	Inclusion
Principle 9. Preliminary anti-action	Pre-Action	
Principle 10. Preliminary action	Pre-Action	
Principle 11. Beforehand cushioning	Pre-Action	
Principle 32. Color changes	Pre-Action	Displace
Principle 28 Mechanics substitution	Inclusion	
Principle 29. Pneumatics and hydraulics	Inclusion	
Principle 38. Strong oxidants	Inclusion	Maximize
Principle 39. Inert atmosphere	Inclusion	Closed cycle
Principle 12. Equipotentiality	Align	
Principle 14. Spheroidality - Curvature	Displace	
Principle 17. Another dimension	Displace	
Principle 30. Flexible shells and thin films	Displace	
Principle 35. Parameter changes	Displace	Inclusion
Principle 13. 'The other way round'	Closed Cycle	Displace
Principle 23. Feedback	Closed Cycle	
Principle 26. Copying	Closed Cycle	Post action
Principle 27. Cheap short-living objects	Open Cycle	Post action
Principle 34. Discard and recover	Closed Cycle	

Several benefits arise from this categorization, when compared to the classic 40-IP:

- The DiP are more logical: they can be developed from both, dialectics and patents, and their abstraction level is similar.
- Exist less overlap: One constant critic to the 40-IP was the overlap between principles, for example principles like dynamics (15), mechanical vibration (18), and periodic action (19) now constitute only one category.
- Easier to remember: Seven principles are easy to remember, and their opposites can be deducted instead of memorized.
- Doing the opposite emerges naturally as an alternative: if a system doesn't work doing something, probably it will work not doing that something.
- Easier to mix: quantity and abstraction level are, again, relevant factors.

In the literature, there are not many references that try to validate or evolve the 40-IP, being more common its reinterpretation in less technical areas, like marketing, sales, advertising, or customer satisfaction (Silverstein et al. 2007). The most elaborated attempt to evolve the 40-IP founded so far is explained in the article "Evolving the Inventive Principles" (Mann 2002). In order to establish a reference point, the approach presented on that article will be analyzed-compared with the dialectical approach proposed.

3.5.3.1. Mann's proposed matrix and the proposed dialectical model

On its article, Mann proposes a matrix constituted by five rows of principles (segment, magnify, re-shape, modify, substitute) and three columns of dimensions (space, time, interface). The contradictions then should be overcome by the intersection of a principle and a dimension, like "segment in time" or "Modify in space" (see Figure 55 and Figure 56). The principles are:

- Segment: segment or merge (i.e. change the number of entities)
- Magnify: make the entities bigger or smaller
- Re-shape: change the external geometry
- Modify: change the internal structure
- Substitute: substitute the existing structure for something else.

It can be observed that the definitions provided are generic, so the user interpretation becomes more relevant.

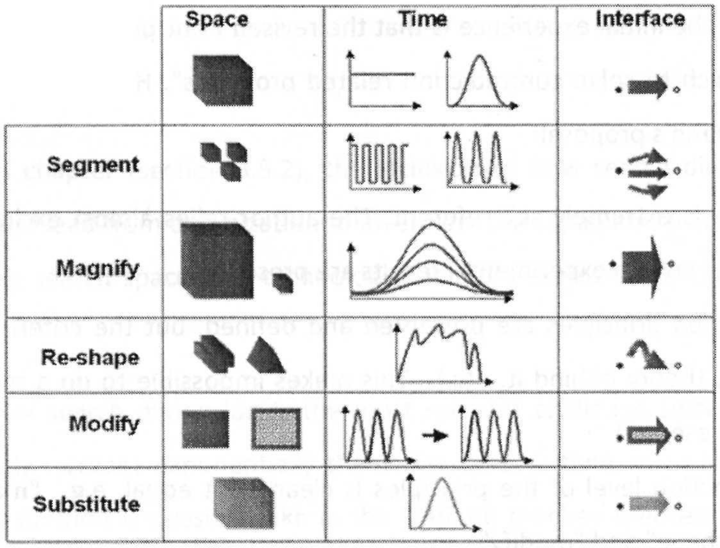

Figure 55: Visual interpretation of Mann's proposal of principles and dimensions.

The equivalence between Mann's matrix and the 40-IP is presented in Figure 56. The following 3 examples are extracted from it:

- Principle 11 beforehand cushioning → Modify in time
- Principle 32 color changes → Modify in time
- Principle 28 mechanics substitution → Substitute in space

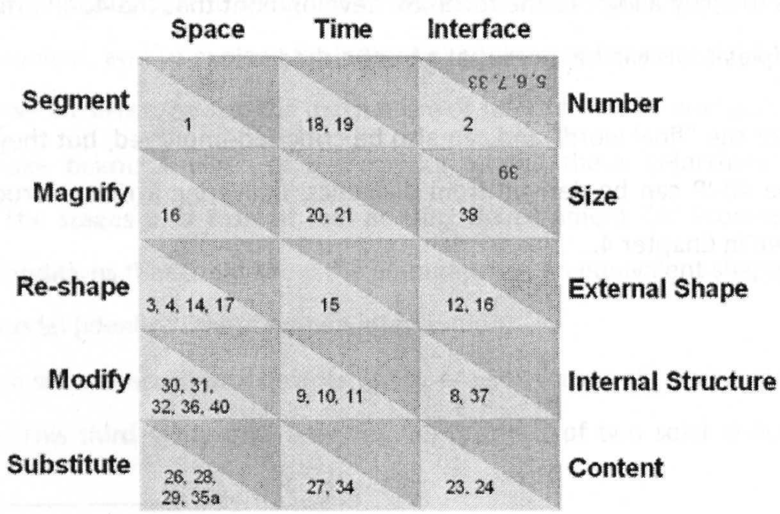

Figure 56: 40-IP inside the matrix of principles and dimensions

It is claimed that "the initial experience is that the revised Principle structure gives a significantly richer structure with which to solve contradiction related problems". However, several observations can be made regarding Mann's proposal:

- The article is extremely self-referent. The author relies almost exclusively on its opinions and experience, and no experimental results are presented.
- The proposed principles are presented and defined, but the criterion to do it is not (utilized technique, theory behind it, etc.). This makes impossible to do a more critical analysis of the material presented.
- The abstraction level of the principles is clearly not equal, e.g. "magnify" is consequence of both, "re-shape" and "modify".
- Similarly, "modify" is so general that can group inventive principles of complete different nature, like "phase transitions" and "composite material".
- The 5x3 matrix presents the possibility of opposites between principles (Figure 56) as alternative to solve contradictions. However, re-shape, modify, and substitute has no opposite in the matrix, and is difficult to think about the existence of them (modify → unmodify?)
- The idea was not developed further. This line of work has not been developed further in other publications of the author.

Returning to the DiP, all the previous observations were considered during its development. At this stage is not considered important if one of the 40-IP has to be moved from category or if it needs a new one, the objective is to apply a logic to the technical development that the 40-IP offers, so its utilization as automation principles-tools can be more that an unfinished idea.

The seven DiP are not the "final word" and can also be criticized-improved, but they are a more robust base from which the 40-IP can be derived from dialectics, delivering a richer structure with practical applications, as shown in Chapter 4.

4. Study cases

4.1. Introduction

As shown in the previous chapter (section 3.5.2), the inclusion of new search dimensions and new evolutionary operators could lead from optimization to invention: the capability to edit first the strategy and then, if necessary, the search space of a technological system seems to be the key to approach automating innovation.

Nevertheless, adding a new search dimension is the most relevant challenge since there are infinite options that could uselessly increase exponentially the computational time. With infinite time and/or power, finding any viable solution is possible (like in the "infinite monkey theorem[13]"), but a realistic approach needs to consider rationally the potential variables (or space dimensions) that will expand the search space versus the available computational resources. In this situation, TRIZ operators have the greater potential of pointing towards new feasible dimensions thanks to their experimental foundations, sustained evolution (official & unofficial), and general versatility . However, it is still necessary to determine (or update) how to measure the quality of a solution. The most common criteria in literature includes setting arbitrary goals or performing multiple runs and applying statistical procedures, but "arbitrary goals are not easy to define either" (Coello 2000) .

Within this context, four study cases are presented as "**proof of concepts",** aiming to validate (and edit if necessary) the viability of different parts of the proposed model:

- Two-Member Truss Design: to show the viability of the dialectical operators in a practical multi-objective problem, and to explore the potential application areas of the DiP. One of them was used inside an EA, inverting also the mechanism of selection of the next generation individuals.
- Railroad brake beam: Analysis of a creativity related thesis (Martinez 1999) in order to determine the stages in a case of real product development CAI Process, together with the particular conditions that could allow the emergence of an equivalent shape proposal within the proposed model (ideally without human intervention).
- Semi-passive solar concentrator devices: In the previous cases, the parameters involved were no intervened. This third study case analyzes the progress of two solar concentrator's proposals

[13] It states that a monkey hitting keys at random on a typewriter keyboard for an infinite amount of time will almost surely type a given text, such as the complete works of Shakespeare. (http://en.wikipedia.org/wiki/Infinite_monkey_theorem).

with the objective of developing the interpretation of the DiP in a CAD context and to determine if this implies new stages in the CAI Process.

- Crash absorption component: It connects different programs executing the whole simulation process for two concepts that aim the same objective. It aims to explore the conditions and limits to computer involvement in the IPS process.

The obtained results are analyzed and discussed in sections 5 and 0.

4.2.Study case 1, two member truss design

In his 2006 article, Kalyanmoy Deb introduces the concept of *innovization* (Deb & Srinivasan 2006), which searches for innovative principles by means of optimization techniques, being the optimization technique used a GA, the NSGAII (Deb, Pratap, et al. 2002b). The idea is that after the multi-objective optimization task has been performed, the set of optimal solutions obtained can indicate the design variables and their objective trade-offs. Those solutions can then be analyzed to determine if there are some common principles among all or many of them.

One of the proposed multi-objective problems of Deb's article, the two-member truss design, is analyzed to explain (and explore) how the DNA can be used to overcome IP conflicts. In this design two objectives are considered:

i. Minimize the total volume of truss members
ii. Minimize the maximum stress developed in both members (AC and BC) due to the application of the 100 kN load.

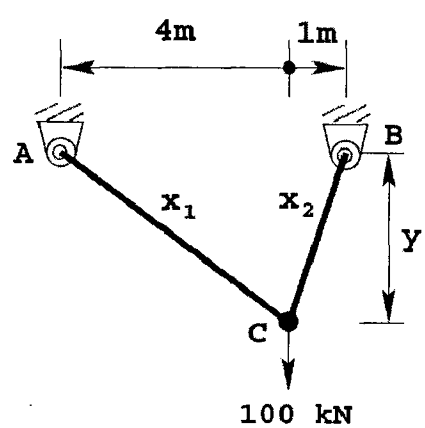

Figure 57: A two-membered truss structure.

There are three decision variables: cross-sectional area AC (x1) and BC (x2) measured in m^2, and the vertical distance between A (or B) and C (y) measured in m. The non-linear optimization problem is given as follows:

$$Minimize \quad f_1(x,y) = x_1\sqrt{16+y^2} + x_2\sqrt{1+y^2}$$

$$Minimize \quad f_2(x,y) = \max(\sigma_{AC}, \sigma_{BC}) \qquad (1)$$

$$Subject\ to \quad \max(\sigma_{AC}, \sigma_{BC}) \leq S_{max}\,,$$

$$0 \leq x_1, x_2 \leq A_{max}\,,$$

$$1 \leq y \leq 3.$$

The stresses are calculated as follows:

$$\sigma_{AC} = \frac{20\sqrt{16+y^2}}{yx_1} \quad (2)$$

$$\sigma_{AC} = \frac{80\sqrt{1+y^2}}{yx_2} \quad (3)$$

The stresses are limited to to S_{max} = 10^5 kPa and the cross-sectional areas to A_{max} = 0.01 m2. For more details please refer to (Deb & Srinivasan 2006).

If Dialectical Principles are to be used as part of a GA, they need an interpretation feasible of being handled by the algorithm. From the principles presented in section 3.5.3, the following ones were interpreted within an evolutionary context:

- **Separation** (Unification): it has been interpreted as the *crossover* operator in EAs.
- **Displace** (Align): it has been interpreted as the *mutation* operator in EAs.
- **Balance** (Unbalance): The balance operator does not have an equivalent in EAs. It consists of selecting a part of the chromosome and giving it a statistical distribution along the population, in this case normal, as the most representative of natural processes in general

4.2.1. Survival of the fittest inverted

In situations when there is no exact criteria to determine what alternative is relatively "best" (as often occurs in inventive processes), it is possible to attack the problems of reducing of the search space and measuring candidate solutions quality doing the selection of individuals the other way around: instead of reproducing the strong (since we do not clearly know what "strong" means), it is better to "kill the weak[14]". For example, if the solution delivered is part of the conventional body of knowledge that does not solve the inventive problem, then the evaluation should be low, and the individuals with uncertain performance should be reproduced. Another reason to utilize the *inverted* selection inside an EA is to discard the possibility that its implementation could block the overall evolution, and to observe its behavior when compared to the traditional selection.

In the spirit of study the "simplest system you think has the properties you are interested in" (Platt 1964), the negation of the system will be the editing of **the optimization strategy** through the concept of dialectics, to discard the possibility that this change has no useful effect at all.

4.2.2. Analysis using NSGA II and VgGA

The algorithm selected to apply the inverted approach is the Virtual gene Genetic Algorithm (VgGA) (Valenzuela-Rendon 2003). As a validation, the problem was also solved using the NSGAII. Similar results were obtained above 100 generations (Figure 58).

[14] This is similar to "steady state", algorithm utilized to avoid the performance decrease during the final stages of evolution (Bentley & Wakefield 1997a).

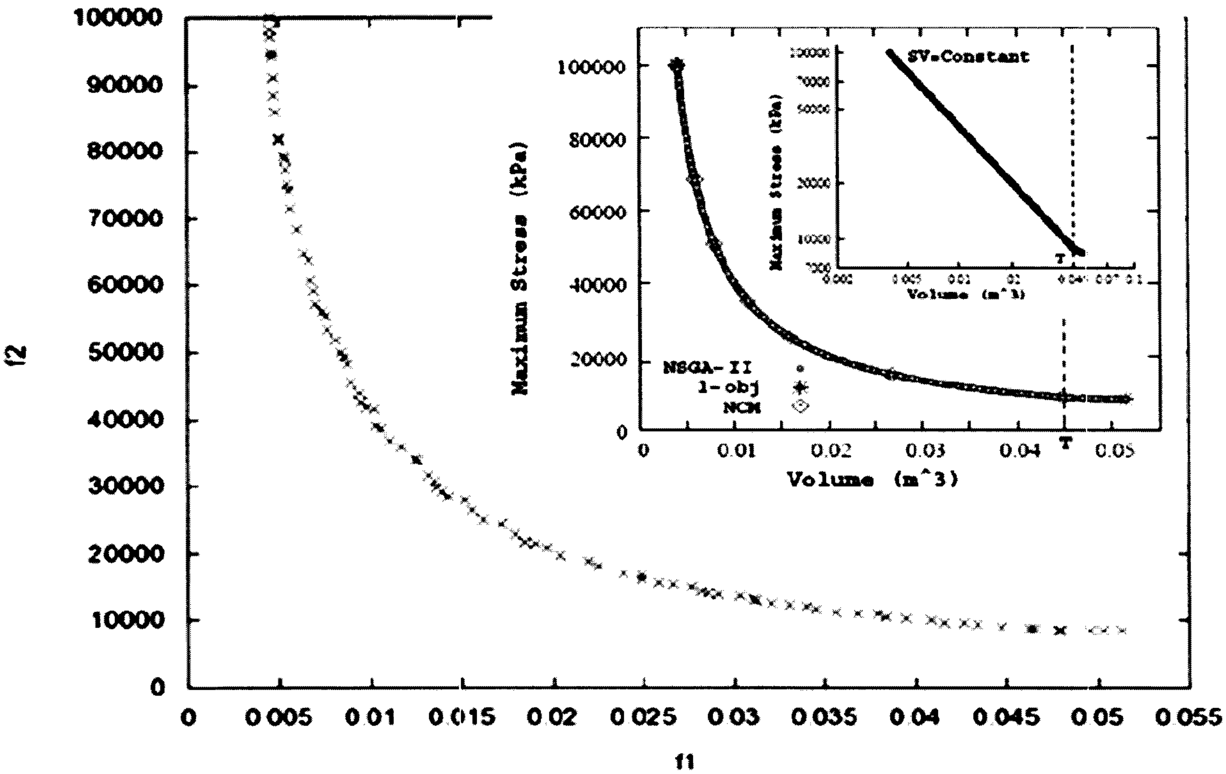

Figure 58: Pareto front obtained using the NSGAII. Above to the right is the graph obtained by K.Deb in (Deb & Srinivasan 2006).

The VgGA is a generalization of a traditional GA with binary linear chromosomes. By mapping traditional crossover and mutation it can simulate linear chromosomes of any integer base, not necessarily binary. Additionally, the VgGA extends where crossover and mutation sites may fall, allowing the simulation of generalized digits.

This generalization has the advantage of versatility, and the extension of crossover and mutation are more compatible with potential requirements of new operators (like balance). The VgGA code was edited to implement a preliminary DNA, using the inverted selection and balance.

In the following graphs (Figure 59 to Figure 63) the principal results are given. The big dots represent the last generation, while the small ones show the values around which the population was moving previously (1000 generations, every 50 printed).

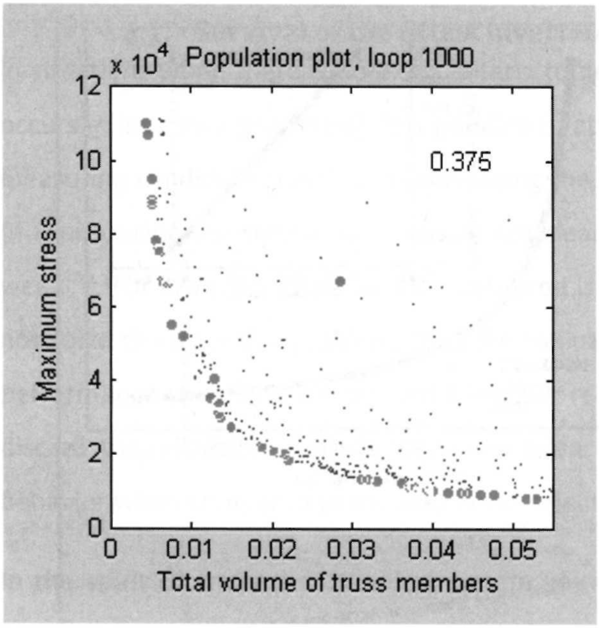

Figure 59: First evaluation using the inverted VgGA. The last generation groups around the Pareto front and the dispersion of the individuals is relatively low.

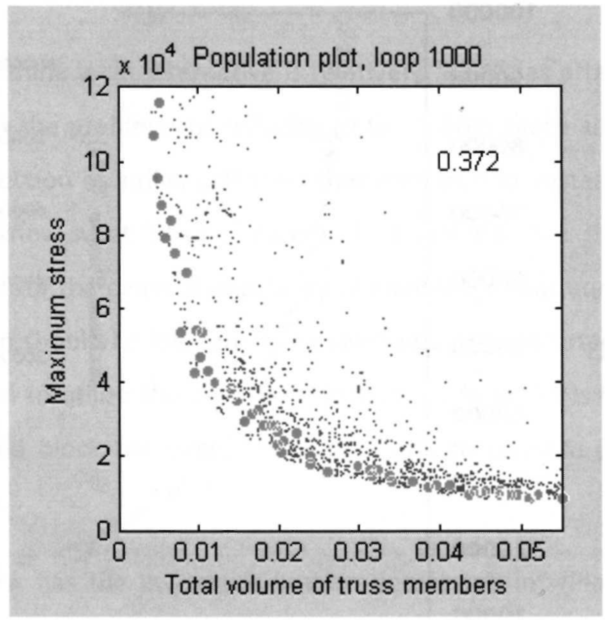

Figure 60: With probabilistic balance, meaning that the algorithm selects at random one variable to be balanced (between x1, x2, y).

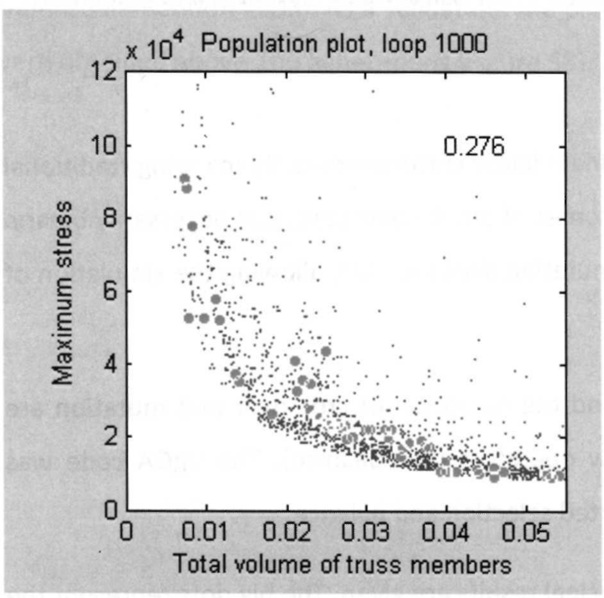

Figure 61: Balance is applied only to x1

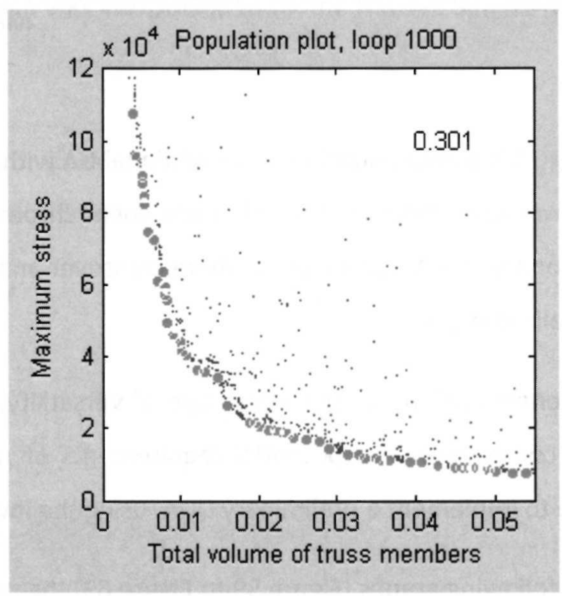

Figure 62: Balance is applied only to y.

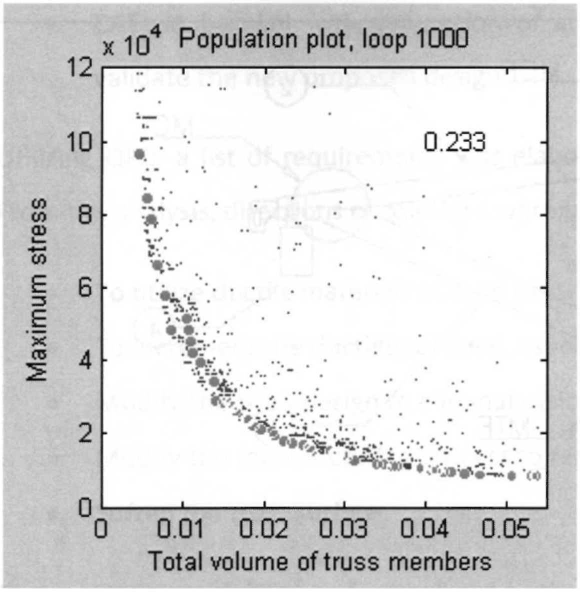

Figure 63: Balance is applied only to x2.

It can be see that the balance operator is far from being neutral. For this particular case, the effect of balance applied directly to x1 (Figure 61) is more notorious, telling us about the relative importance of the different variables involved. The analysis of the potential consequences is presented in section 5.2.1.

4.3.Study case 2: A railroad brake beam

In his thesis "Integration of innovative design methodologies and parametric optimization applied in the design of a rail road brake beam[15]", Martinez analyzes the generation of innovative design by integrating several methodologies, in order to obtain products with better performances.

The developed study case, about how to modify the parameters of a railroad brake beam (Figure 64), is used as a reference to clarify:

- The stages to consider in real (manufactured) CAI Process
- The necessary conditions that could allow the emergence of an equivalent design proposal (valid solution to the problem) within a more automated process, ideally without human intervention.
- If new stages need to be developed in the proposed DNA.

[15] Originally in Spanish: "Integración de metodologías de diseño innovativo y de optimización paramétrica aplicadas en el diseño de una viga de frenado de ferrocarril" (Martinez 1999)

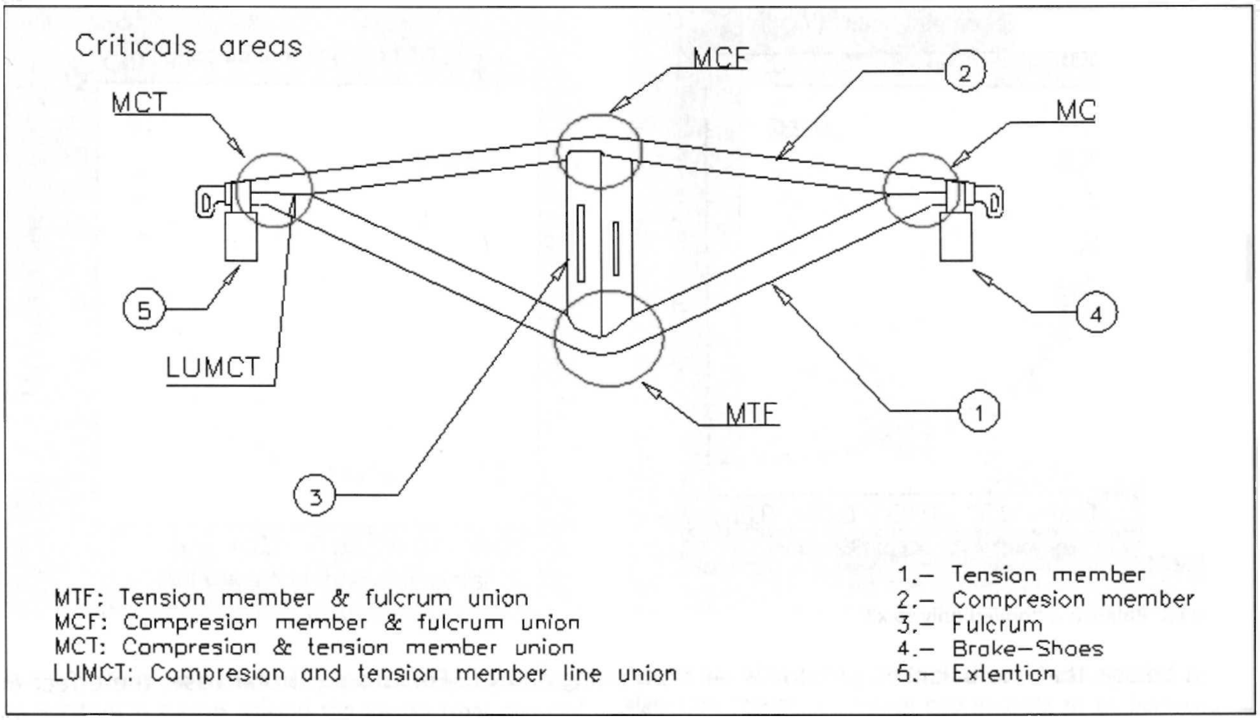

Figure 64: Diagram showing the brake beam basic elements, and the critical areas needing improvement.

In the study case, the performance of the brake beam has reached its Pareto front without giving the desired results, being necessary to look for alternative solutions. Depending on design decisions, this can be considered a level 2 problematic (change the search strategy) or a level 3 one (change the parameters considered).

4.3.1. Development using QTC methodology

The general objective is to obtain a new shape with the desired performance, reducing manufacturing costs as also structural weight and stress. The utilized solution process follows the QTC methodology (QFD+TRIZ+CAD) as follows:

- Quality function Deployment (QFD): to determine the internal requirements (cost, weight, security, others)
- TRIZ: to the conceptual generation of alternatives, based on the 40-IP and SuH diagrams to study the interrelations between functions, assisted by the software IWB. This is the more substantial part of the new product development.

- CAE: in parallel with generation of alternatives, first to model and finally to simulate and validate the new proposed design.

Utilizing QFD, a list of requirements was elaborated, to later establish its relations using TRIZ tools[16]. From the analysis, directions of solutions were proposed:

- To utilize ductile materials instead of steel.
- To increment the ductility of steel, especially in places where fissures appear.
- Modify the truss design to eliminate (sic) the stress concentration.
- Modify the manufacturing process to reduce the stress concentration.
- Soften the truss surface.

The stress reduction direction is analyzed further, being suggested:

- Change the positioning angle of the beam.
- Avoid unbalanced loads.
- Modify the truss transversal section
- Create a pre-tension in the beam design to increment the stress and reduce weight.

The final decision criterion is the local impact of the selected alternative: it must not interfere with the brake system, only with the brake beam. Considering it, the conclusion is that the better alternative is to redesign the truss, changing its transversal section using a commercial available alternative. The most viable components to apply this change are the tension and the compression members (Figure 64). The new shape is presented in Figure 65, on which the section of the tension member is now circular, implying also the redesign of the fulcrum.

[16] Specifically the SUH diagrams. They were developed by Ideation International (http://www.ideationtriz.com/), and are considered part of the "modern TRIZ".

Figure 65: New design based on a tube as the tension member.

The identified necessary stages during the CAD assisted process consisted basically in:

1. Original CAD Model Development.
2. CAD export
3. Meshing
4. Analysis
5. Evaluation (end if objective is accomplished)
6. Selection (by the user-operator)
7. Edition of parameters (by the user-operator)
8. Return to 2.

In this particular case, the critical change was during the edition of parameters: new truss profiles were evaluated from commercial alternatives, being selected the 2" tube. When compared with the initial brake beam, the new version was a 5% lighter and obtained a 23 % stress reduction.

4.4.Study case 3: Semi-passive solar concentrators

In section 4.3 case study, there are several characteristic unmodified, like the number or relations between components, material involved, the railroad, etc. In this new study case it is analyzed the development of a solar concentrator (Figure 66), following two conceptual branches that expand the standard solution search space. The objectives of this third case are:

- Develop the interpretation of the dialectical operators in order to be useful within a CAI context.
- Determine (clarify) the necessary conditions to propose and develop, by computationally means, different alternatives to solve a particular problem.

It is intended to go beyond the modification of the optimization strategy involved and modify the search space changing the parameters involved (range or quantity). This is a level 3 situation that can be turned into a level 4 one, since there are many initial assumptions (conditions) that could be decided to modify after the study of the physical principles involved, thus changing the problem to be solved.

Figure 66: The main components of a typical solar concentrator are the parabolic concentrator (1), the receiver (2), the following system (3), and the supporting structure (4). The initial position of the tracking system should be calibrated to the particular position on the Earth's surface due to latitude variation.

The two conceptual proposals (still on development) analyzed are:

- Proposal 1: Alan Anaya "Optimization of a Fresnel lens using Genetic Algorithms for solar thermal concentration and tracking purposes"
- Proposal 2: Carlos Ramirez "High efficiency solar concentration system with minimum solar trajectory following[17]"

4.4.1. Optimization of a Fresnel lens using GAs for solar thermal concentration and tracking

The particular objective is to design a solar concentrator able to achieve 1000°C without following the sun (passive). Ideally, it should not have any mobile mechanism, and the radiation must be focused into a minimum area throughout the day.

To accomplish this target a compound Fresnel lens has been selected to be developed (Figure 67). This is far from being trivial because solar constantly change their incidence angle, which is also related with the refraction/reflection (through Snell's Law[18]).

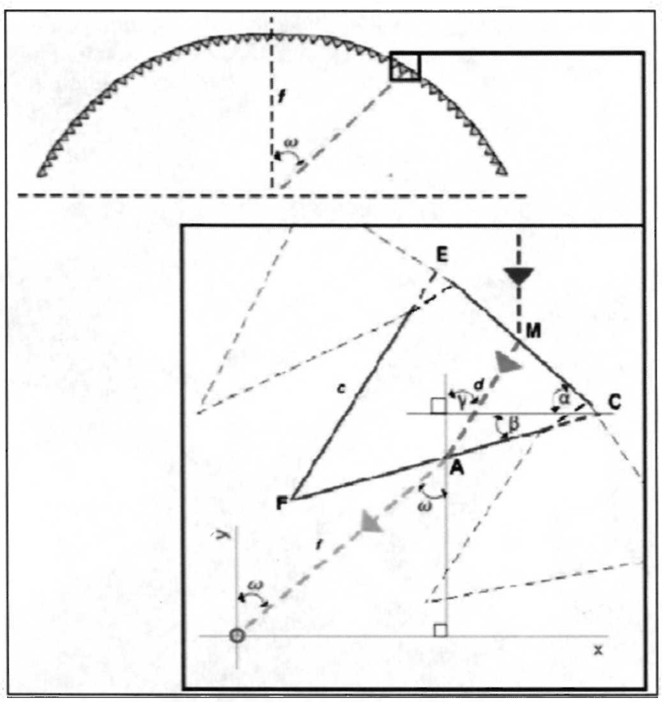

Figure 67: Fresnel lens profile diagram, necessary to construct (calculate) the prism's angles and position that deliver the intended focal point.

[17] Originally on Spanish: "Sistema de concentración solar de alta eficiencia con seguimiento mínimo de la trayectoria del Sol"

[18] Its explanation is available in physics books (e.g. (Serway et al. 2000)) or the internet (http://en.wikipedia.org/wiki/Snell's_law).

The ideal final result of a fixed solar tracking and concentration device is blocked by the conflict between concentration capacity and complexity of the collector. This contradiction guided the decision of using TRIZ contradiction matrix to found useful inventive principles. In this case, the parameters in contradiction are *Illumination Intensity* and *Device Complexity*, being the solution proposed the inventive principle *6.Universality,* which states "make an object to perform multiple functions, eliminate the need of other parts".

The interpretation consists in to obtain a geometric shape of the lens able to refract all incident rays towards the same spot, regardless the incidence angle. Once the conceptual proposal has been defined the process is automated as much as possible inside an evolutionary cycle (see Figure 68), being the critical step the optical simulation, that allowed the selection of individuals.

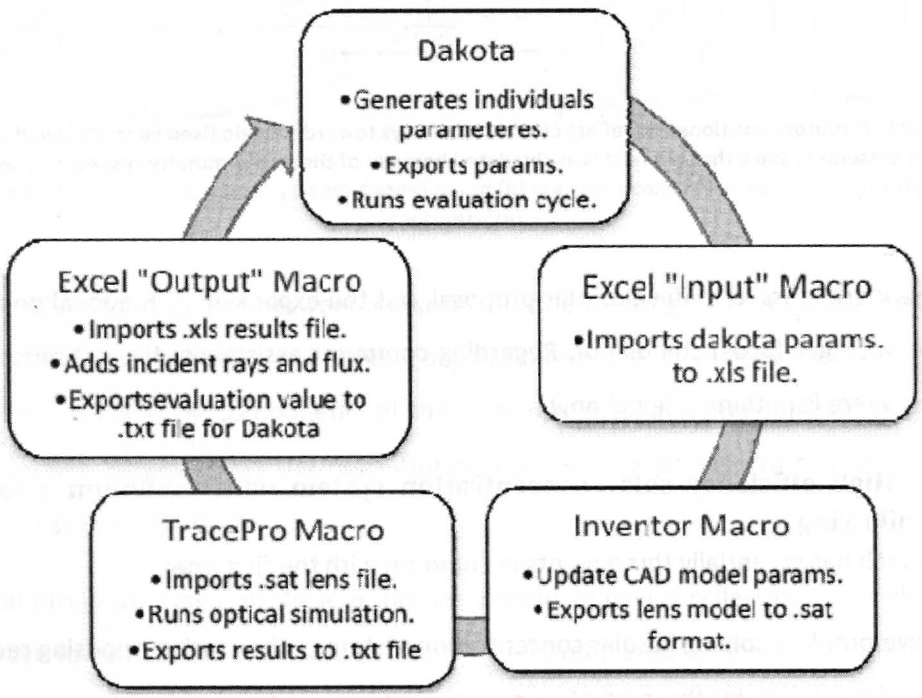

Figure 68: Descriptive diagram of the developed optimization process.

The obtained results first showed improvements in the global efficiency (along the day) from 12 to 19%, which encouraged the introduction of new receptors in critical spots with the capacity to refocus the received rays towards a new common point of concentration (Figure 69). The estimated efficiency of this new idea is a 52%.

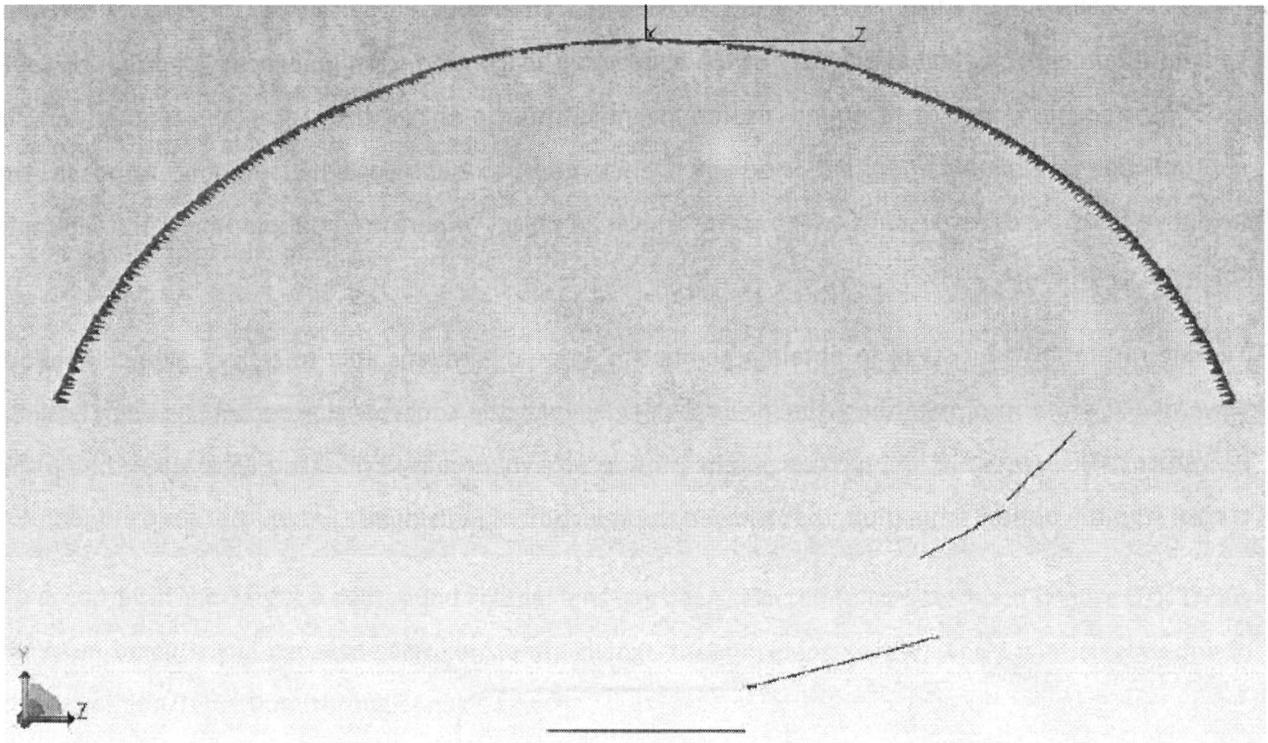

Figure 69: Multiple receptors positioned to reflect concentrated rays toward a single fixed spot. Only half of the receptors are presented, since that is how it was simulated because of the lens symmetry respect to y axis.

Several barriers were crossed to develop this proposal, but the expansion of the focal points in quantity instead of size (a bigger focus) stands out. Regarding computer assistance, the optimization process is automated, but there is nothing original on it.

4.4.2. High efficiency solar concentration system with minimum solar trajectory following

This second branch has essentially three points in common with the first one:

- Inventive problem solving in solar concentration systems, when facing opposing requirements.
- The conflict caused by the cost of a tracking system and the need of a constantly controlled movement.
- The proposal development involving the use of computer simulation, which leads through a series of optimizations and validations.

Within solar concentrators, there are basically two types: two-dimensional, and three-dimensional. The first one concentrates the solar energy throughout a line, while the latter does it in a spot. In this case the concept is to be developed as a three-dimensional, more specifically the so called Dish/Motor (DE), presented in Figure 70.

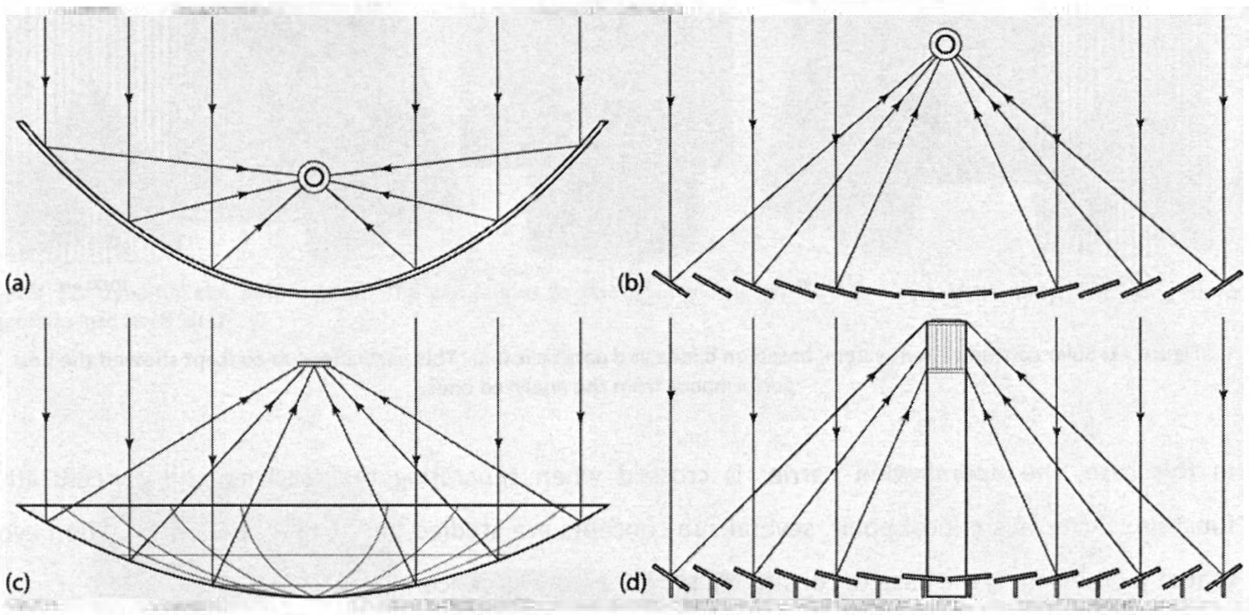

Figure 70: Typical solar concentration systems. Cases (a) and (b) are linear concentrators, while (c) and (d) are spot concentrators.

The conflict in this second case is between *adaptability* (the tracking system deliver that) and *energy*, being the recommended inventive principles *1.Segmentation* and *19.Periodic action*. The first one is interpreted as to separate the components of the system that follows and concentrate the solar rays. The second principle suggests that the repositioning can be periodic instead of constant, in order to diminish the energy consumption.

The conceptual proposal, based on those principles, consist in to use reflective blinds that re-direction the incident rays towards a static parabolic disc, which immediately concentrates the rays towards a single focus. The adjustment in the position of blinds and disc is made every hour. Figure 71 illustrates the concept.

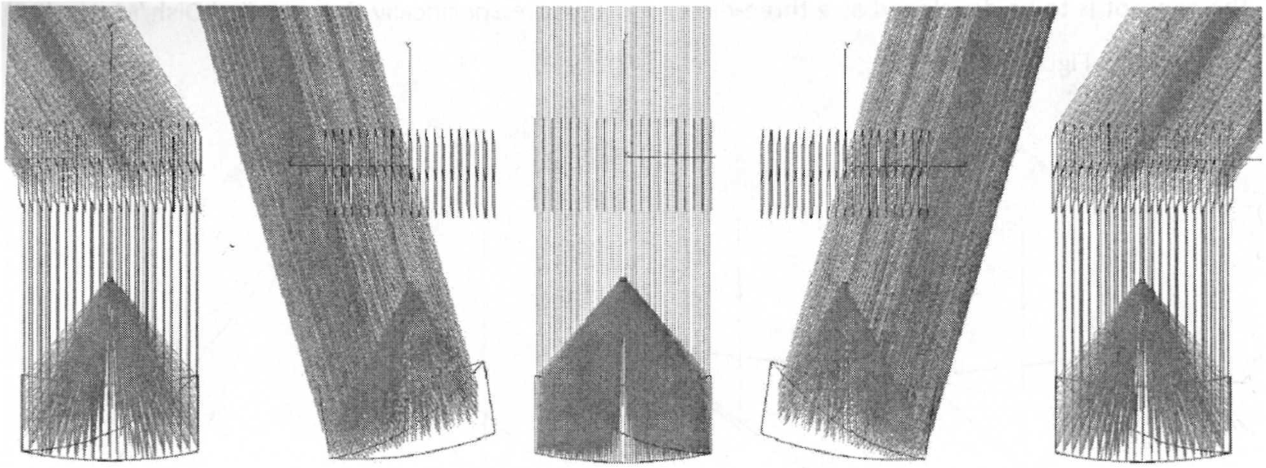

Figure 71: Solar concentration system, based on blinds and parabolic lens. This particular sub-concept showed the best performance from the analyzed ones.

In this case, the optimization barrier is crossed when separating the tracking and concentration functions. After this critical point, several sub-concepts are studied initiating a new optimization cycle that at the present is the base to construct a prototype of the concentrator.

4.5.Study case 4, Crash absorption component

In his thesis "Structural optimization of a crash absorption component", Sewe optimizes and evolve a crash absorption component within an automatic loop, by connecting different programs and executing the whole simulation process. Its general objective is to explore proposals faster with less work (Sewe 2008).

Initially, relevant observations are made comparing the model of a commercial can with a real one: the behavior is notoriously different, probably due to the multiple forces applied to a can during its life (transport compression, material irregularities, opening, others), meanwhile the simulation can is "perfect" (Figure 72 and Figure 73). This shows the necessity of revising carefully all simulations' results.

Figure 72: Dynamic can deformation. The can begins to buckle in one unpredictable point of its body, collapsing more regularly into itself later.

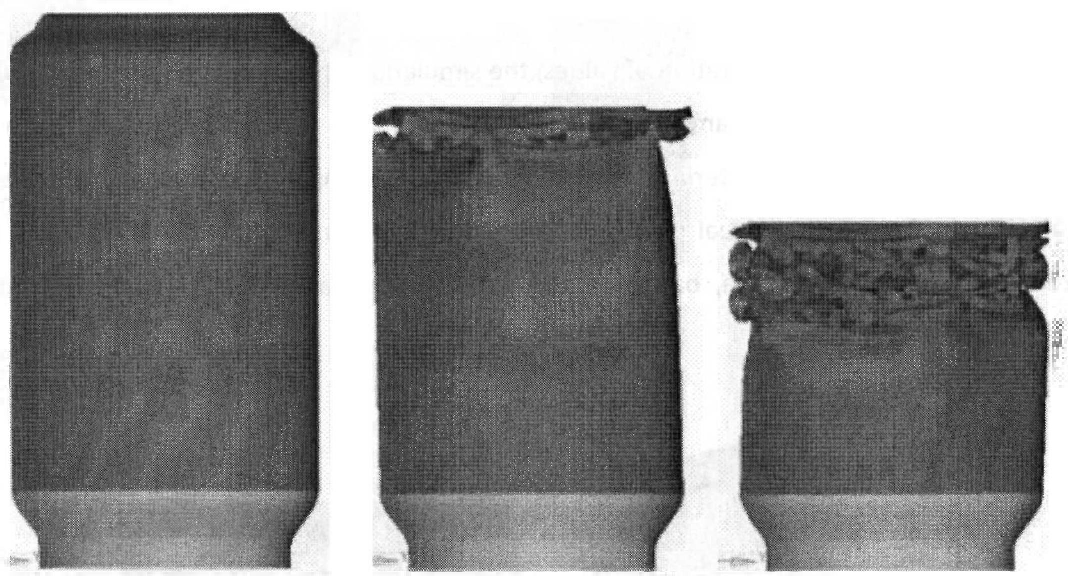

Figure 73: Simulation of the dynamic can deformation. It can be seen its regularity all along the compression of the body.

The development of the crash absorption component starts under the hypothesis that the component would absorb more energy when a force is applied longitudinally and the element thins at the middle, allowing one section to insert into the other. The **first concept** was developed testing paper models (Figure 74), to later explore the search space via simulation.

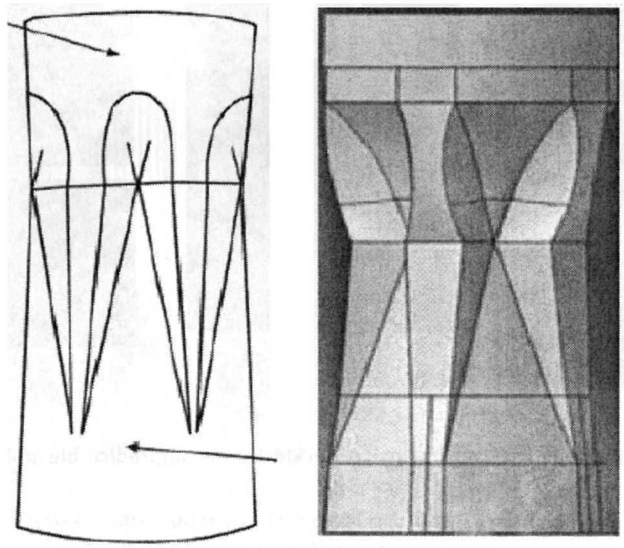

Figure 74: First ideas sketch (left), and CAD model prior optimization (right).

In order to find the optimal configuration of values, the simulation process was automated using a loop, starting with the creation of a parameter set that ended with the result analysis. Utilizing a GA, the shape was optimized under the criteria of deformation time: a longer time to collapse implied more energy absorbed. The best individual (Figure 75) obtained a deformation time of 57.73 ms, and the worst 28.62ms. The original shape, based on the paper models and intuitive modifications, had a deformation time of 49.67ms.

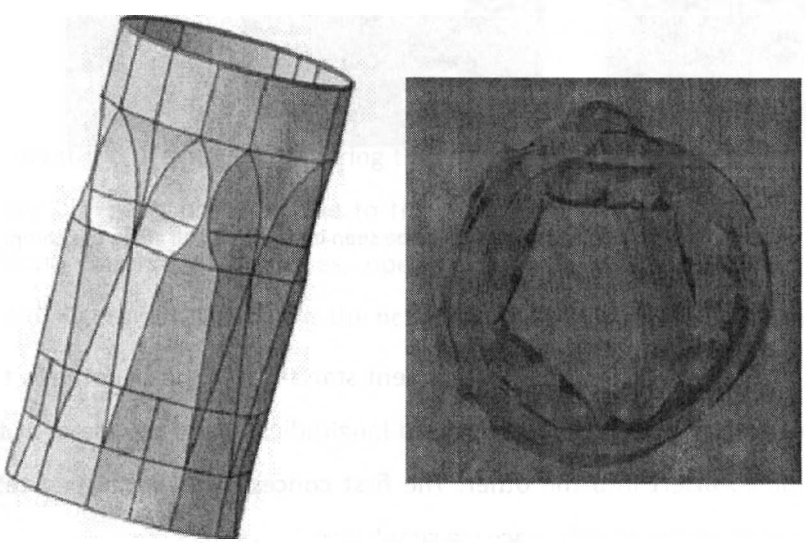

Figure 75: Best performance geometry, general view (left) and top view (right). Both results are distant from the original expectations (intuitive ones).

Despite the improvement, the question about if there is another better geometry remained unanswered. To explore that area, new ways to obtain shapes were discussed and another "optimization loop" was developed. This **second concept** utilized a relatively simpler element to verify the loop, looking forward to apply it in more complicated geometries. The central idea is to take advantage of the rigidness of a tube, but softening the high force in the beginning by controlling it using the diameters of the included holes (Figure 77).

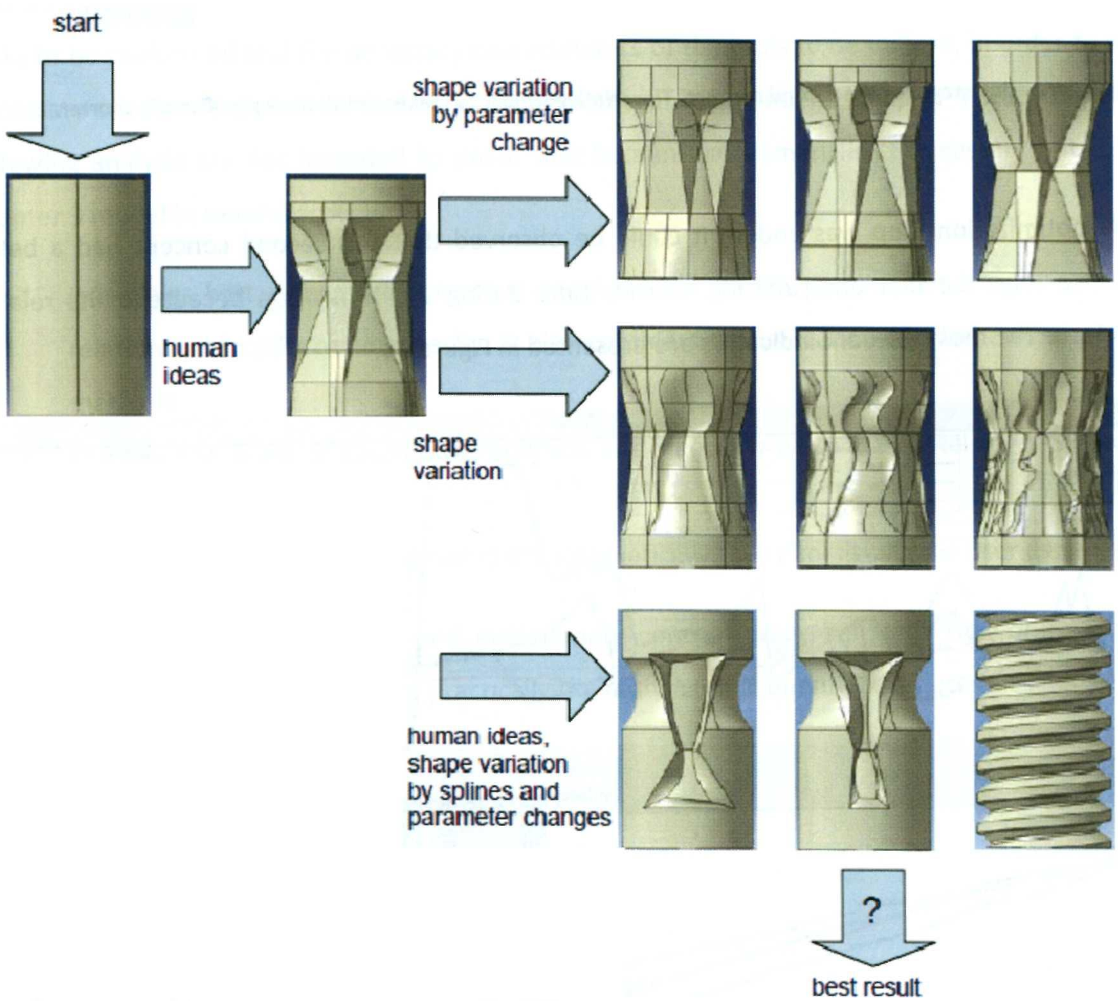

Figure 76: General view of the followed optimization process. Once the limits of the parameters are determined the improvement can be carried out automatically (first row) or manually. The "screw" idea could not be simulated due to meshing characteristics.

Figure 77: Deformation stages of the second concept. The relative simplicity makes simulations significantly shorter.

After the optimization loop was ended, it could be observed that this second concept had a better performance than the first one, utilizing also les time during the simulation because of its relative simpler mesh. The most relevant indicators are presented in Figure 78.

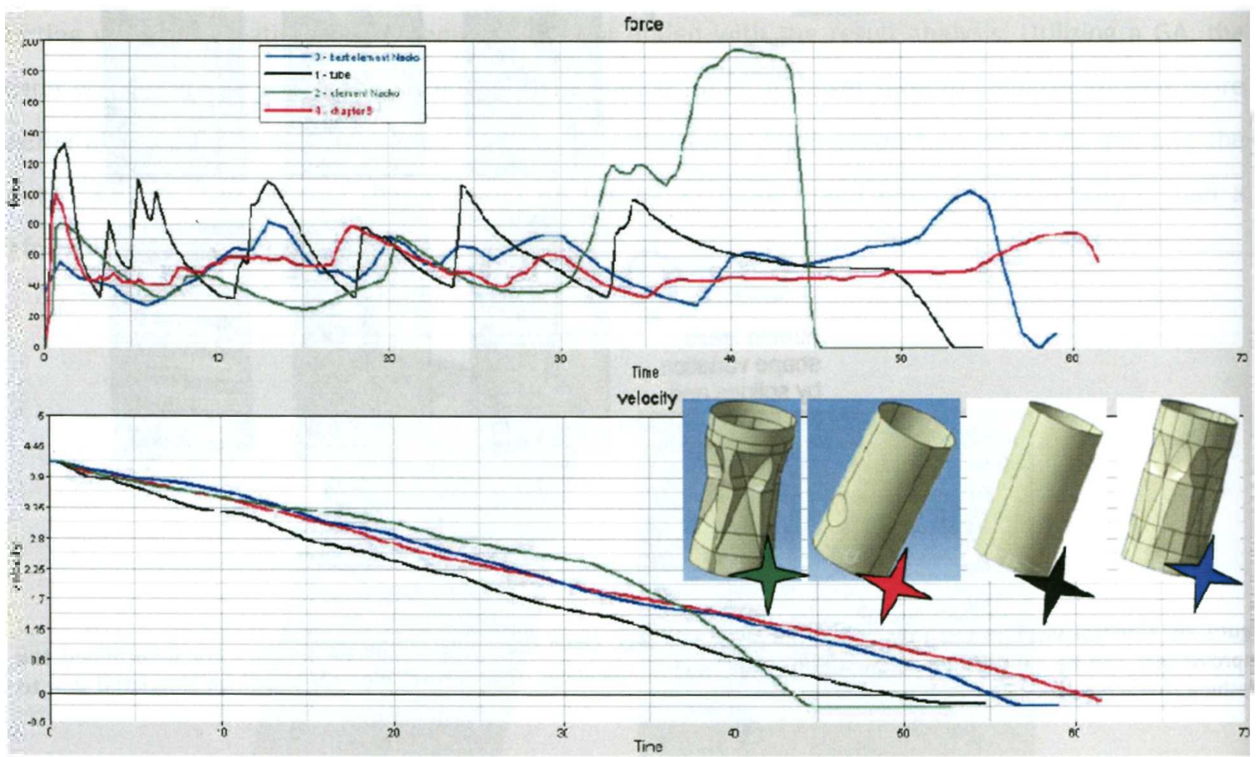

Figure 78: General performances of relevant shapes, from left to right: the first concept, the second concept optimized, the tube, and the first concept optimized. The second concept optimized had a longer time to collapse (inferior graph), which is consistent with its more uniform force distribution (superior graph).

5. Obtained results and analysis

This section presents the analysis of chapter 4 study cases, under the perspective of chapter's 3 proposed model.

5.1.Restrictions

This dissertation develops a procedure to solve IP based on dialectical negation, by describing the critical tasks to be performed and the necessary characteristics of the tools to be utilized, in order to study the necessary conditions that could make an IPS method user independent. Chapter 4 study cases and the following analysis are not intended to prove that human involvement isn't necessary but to validate chapter 3 model in several aspects:

1. It explains better than the current *state of the art* the IPS process, meaning its components, relations and cyclic nature mostly. It also considers the utilization of trial&error and structure methods.
2. Clarifies the need of considering *conflicts* and *negation* as fundamental for IPS, showing its mutual dependence.
3. It can be used as a guide to develop IPS automatically thus diminishing the user dependence.

Dialectics is the main framework, and evolutionary concepts together with TRIZ principles are the applied methods to implement more practically the model. Consequently, they are not under evaluation or comparison.

5.2.Analysis of the study cases

5.2.1. From the study case 1 (Section 4.2)

The first study case was made using two distinctive characteristics, *inverted selection* and *balance*, in order to show that a new operator (*balance*) can deliver oriented results going beyond traditional mutation, and the possible results of negating a system through changing the search strategy (*inverted selection*).

The *inverted selection* approach arrived to similar results that the standard procedure, but it was significantly slower (more generations were required to reach the Pareto front). This was expected since the mediocre population was reproduced, "destroying" the learning of the GA while maintaining the diversity without explicit intervention (like utilizing *injection* (Coello 2000) or another GA technique). The previous has the positive side that, whenever required, the dispersion of the sample can be intentionally increased.

Regarding *balance*, it had a moderate effect increasing the dispersion of the sample when it was applied to y and x_2, and a relatively intense effect when it was applied to x_1 (Figure 59 to Figure 60). It was expected that the normalization of a part of the chromosome could significantly affect the learning of the GA increasing the dispersion, but the fact that this happened mainly with one of the variables is a surprising behavior. The underlying reason for this might be evident to an experienced mechanical designer (the arm length that produces the torque of x_1 is four times longer than the x_2), but *balance* makes this characteristic more tangible to anyone who looks at the figures, no matter what their technical background. It is reasonable to think that *balance* can be used to attain several objectives:

- To increase the dispersion of the sample, and lead to alternatives in situations when the objective is not completely defined, or the optimal solutions (Pareto front) cannot be used for some particular reason (cost, downtime, others).
- As a way to measure the relative importance of the variables, by showing its effects on the output (like x_1 in the study case).

The case study suggests that with the right operators, the candidate solutions could be directed towards the users' necessities and objectives, instead of being limited by its partial knowledge about the involved parameters and restrictions (search space). The slower results obtained through *inverted selection* are a clear disadvantage for optimization problems, but the case presented does not aim to optimize, neither this dissertation.

Regarding the human involvement it is constant until the optimization cycle starts: the invention here is not in the tools, but on its arrangement and interaction.

In general, it can be said that this first study case shows that the utilization of a DiP can produce discernible effects, validating the viability of the dialectical operators in a practical multi-objective problem. This is a fundamental first step to explore the potential application areas of the principles by themselves or incorporated into EAs.

5.2.2. From the study case 2 (Section 4.3).

In the first case study only the *optimization strategy* was edited. To test the editing of the *search space* and the *assumptions* (see Figure 51) several challenges arise, starting with the criteria by which modifications should be made, followed by the computational problems that the integration of bigger search spaces create. To solve the *brake beam* problematic a new truss profile was introduced from commercially available alternatives. These alternatives, that finally allowed crossing the optimization/invention barrier, were part of the designers' knowledge, and in no case the result (direct at least) of utilizing CS during the process: computational tools were useful once the concept was already decided.

Regarding the generation of alternatives, QFD and TRIZ were used to elaborate a list of requirements and then directions of solutions (section 4.3.1), being selected the "stress reduction" as the most viable to be analyzed one step further. For each one of the alternatives, it was possible to found a DiP that could suggest the same idea (Table 10), showing the potential utility of DiP in practical situations.

Table 10: originally proposed development direction and the DiP with the potential of generating it.

Original proposed direction	Dialectical Principle (section 3.5.3)
a) To utilize ductile materials instead of steel	Closed cycle (Open cycle): In this case the cycle is opened to new materials.
b) To increment the ductility of steel, especially in places where fissures appear	Minimize range (Maximize): the property of ductility is maximized.
c) Modify the truss design to eliminate (sic) the stress concentration	Pre-action (Post-action): the problem is solved by design.
d) Modify the manufacturing process to reduce the stress concentration	Pre-action (Post-action): the problem is solved by manufacture.
e) Soften the truss surface	Align (Displace): to make homogeneous the surface of the truss.
f) Change the positioning angle of the beam	Pre-action (Post-action): the problem is solved by design.
g) Avoid unbalanced loads.	Balance (Unbalance): to distributes uniformly the loads along.
h) Modify the truss transversal section	Pre-action (Post-action): the problem is solved by design.

Analyzing the process followed by Martinez to solve the problematic helped to clarify several points, standing out:

1. The determination of the stages utilized in a real CAI process (already presented in section 4.3.1)
2. The verification that the steps of those stages are coherent with the steps proposed within the DNA.
3. The particular conditions that could allow the emergence of an equivalent shape proposal within a more automated process, based on the DNA.

The second point of coherence can be appreciated following Figure 51: CAD is the tool used to negate the actual system, in order to use simulation to analyze and reach a logical status, which will finally allows selecting proposals to evaluate.

The third point requires to clarify how to modify the original search space of a problem, meaning the geometry of candidates. Following the brake beam, in order to modify the geometry in a more automatic context, the CAD model:

- Needs to be parametric
- Sketch-based
- Able to be restricted (e.g. maximum dimensions)
- Able to be reconstructed after a parameter is modified.

More particular requirements for those steps are presented in Figure 79.

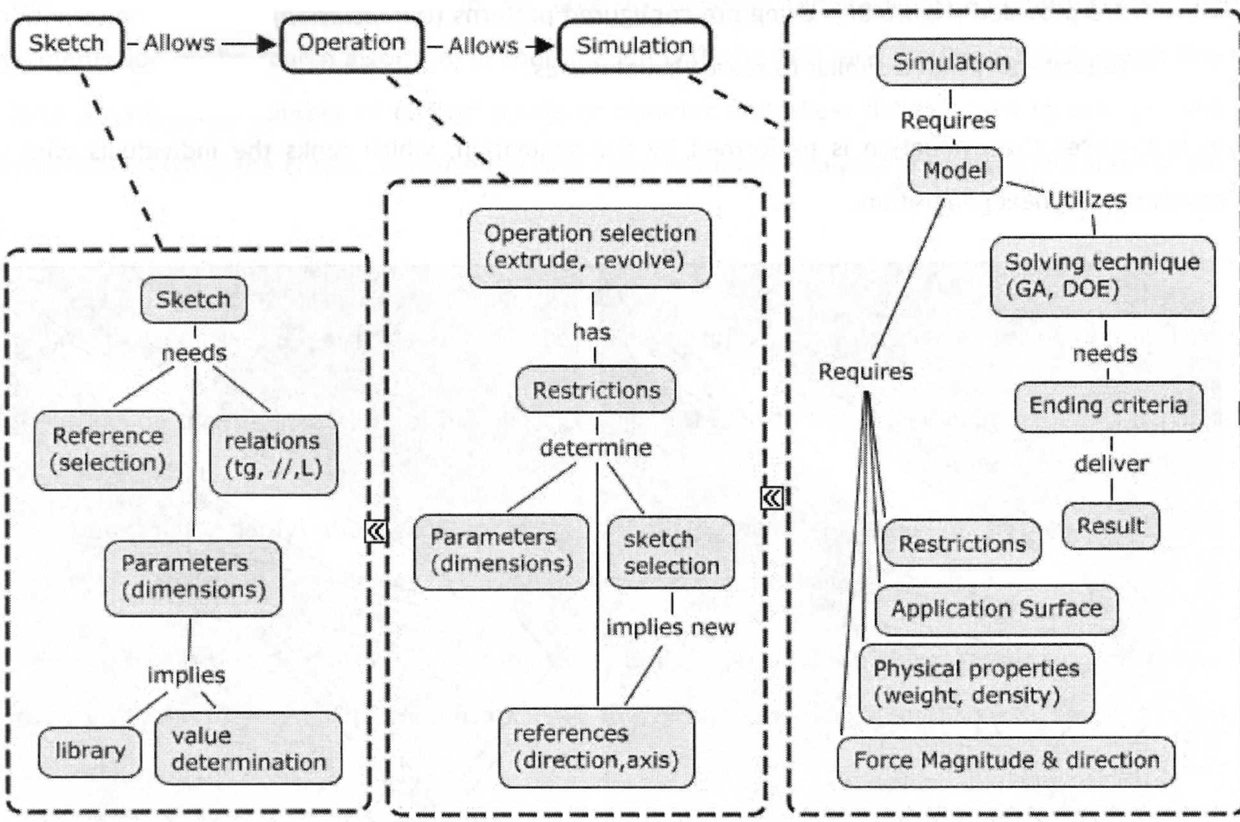

Figure 79: The identified necessary stages in a CAD process that aims to improve-innovate a product. This work is still under development, but it points out relevant characteristics to consider when modifying a CAD model.

Contrary to study case 1, CS is a secondary element participating in the mechanical analysis. The selected concept, consisting in a beam truss redesign changing its transversal section using a commercial alternative, can be replicated by a more automated system in at least two ways:

- The shape search can be done first in available materials databases, not being necessary modifying every parameter, similarly to how a morphological matrix works. Implementations of similar ideas are now a common practice[19] once there is a decision about the component to modify and its major limitations.
- If the sketch is constructed utilizing several control points, EAs can explore the pre-defined search space of each component. The shape variations induced by the reproduction of genes or more specific DiP (like *Align*) has the capacity of deliver the same results if enough control points

[19] There are many CAD and catalog developers partnerships, like PTC and Traceparts : http://www.ptc.com/appserver/wcms/partners/software/company.jsp?&im_dbkey=22412&icg_dbkey=844

are provided (Figure 80). Using pre-configured patterns (DiP *Inclusion*) is another alternative in this category that is similar to search in databases.

In both cases the evaluation is performed by the simulation, which ranks the individuals who will constitute the next generation.

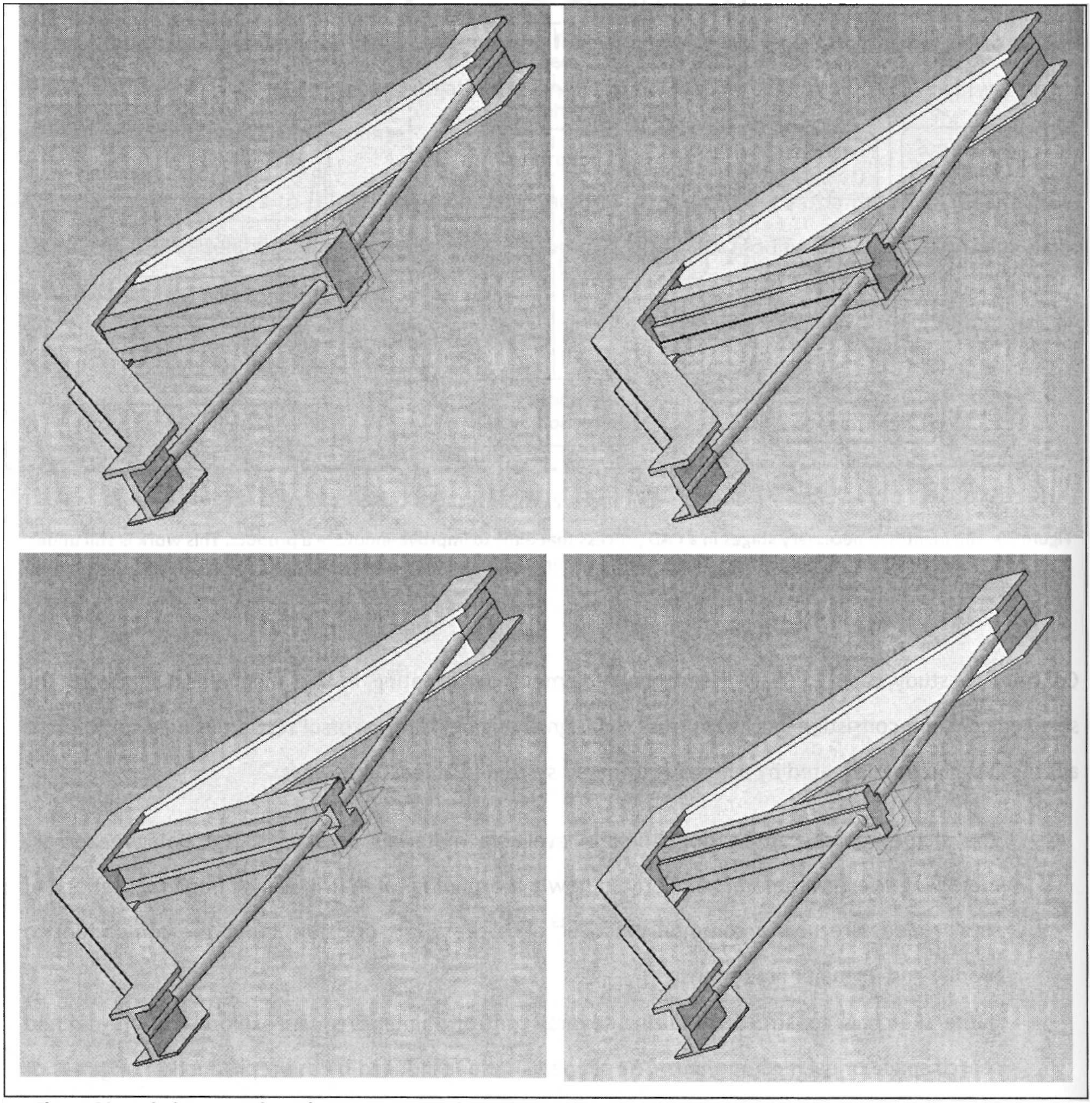

Figure 80: variations on Fulcrum's parameters without altering the relations between components. The section can easily include more control points, able to deliver an approximate circle if that is the proposed solution.

In brief, the essential requirement to be able to develop this concept automatically is the implementation of the search algorithm in coordination with the CAD model edition, which needs first to have an adequate number of control points or libraries that allow the relations to emerge. The coordination stage is the critical, because the other two (sketch & analysis) are part of the state of the art.

5.2.3. From the study case 3 (Section 4.4)

This third study analyzes the progress of two solar concentrator's proposals, in order to:

- Develop the interpretation of the dialectical operators, with the objective of use them within a CAI context.
- Determine (clarify) the necessary conditions to propose and develop, by computationally means, different alternatives to solve a particular problem.

The main difference with the previous study cases is that the concentrators are studied, in relative terms, as problems where the modification of the parameters and geometries involved has no predefined boundaries.

Each case is being solved independently, but based on common principles: after the decision of working with concentrators, a contradiction was found and studied, reaching inventive principles that aided the generation of alternatives to later validate and optimize the concept in a computational environment. The proposal of the alternatives was done using intuition and technical knowledge (expertise) in different ways: in case 1 was automated after the concept was developed enough to be put inside an optimization process, and in case 2 it was mathematically modeled after some experimentation that supported the basic idea.

The creative stages during the development of the Fresnel lens (Section 4.4.1) were performed by designers. After the analysis of contradiction lead to the inventive principle of "universality", a CAD shape was utilized to generate a GA optimization loop, which helped the designers to come up with new ideas. The DiP of *Minimize range (Maximize)* has the same potential than the used one, being one of the most direct equivalences.

In the case of the minimum following tracking system (Section 4.4.2) the main conflict "adaptability v/s energy" lead to cross the optimization barrier when the functions of tracking and concentrate rays are physically separated, applying the recommended *segmentation* and *periodic actions*. From that point the generation of ideas is done by the designers without following a method. No EAs are utilized to optimize, but mathematical and CAE models to evaluate. Regarding the DiP's, in this case *Separation (Unification)* is the one that offers the equivalence of both TRIZ principles utilized (a periodic action is a segmentation in time).

One important thing to note is that in both cases, **it is necessary to modify the relation between parameters**, not only its values, being this the most probable reason why the conceptual stage is not assisted. The creative leap cannot be replicated by computers in a static search space, and the solar concentrators are a good example of that. The definition of restrictions is the guide that defines the problem, and the intended innovation can be achieved only by the ability to handle it beyond randomness. Maybe simpler problems could allow just having relaxed restrictions, but real life ones (like the presented) need a clear definition of the problem limits and search directions. At least in the concentrators, the team's proposals cannot be replicated by a computational system.

Visualizing how to achieve this editing capacity by computationally means, some fundamental elements must be maintained:

- Fixed source (sun)
- Objective (evaluation function)
- Active principle (concentration)

The preceding can be used to define an editing logic when results are not acceptable, from superficial modifications (e.g. parts dimensions) to fundamental ones (e.g. energy source). Inside this process, the DiP can be sequenced (even defined) to follow the criteria to edit both, the search space and the search operators. All the previous remark the importance of the ideal result definition and thus the **objective function:** if the objective is changed, the system as a whole is destroyed.

Returning to the stages performed by designers, it is observed that the overcoming of the contradiction (by *universality* or *segmentation* in the cases) is the critical stage not aided by the automation, confirming that intuition and expertise are the obstacles to surpass.

In order to establish how the search for alternatives can be assisted within a CAD context, the DiP are reformulated (Table 11) taking into account the study cases observations.

Table 11: DiP within a CAD context.

1. **Separation** (Unification): The different elements of a system are made more independent. In a CAD context it means to separate operations.
2. **Minimize range** (Maximize): Affects the extension of the actions applied to a system (partial, general, concentrated, diluted, etc). CAD: should affect the extent of operations, so its references (endpoint, tangent, etc).
3. **Balance** (Unbalance): it distributes uniformly the components or actions within a system. CAD: Symmetry in the distribution of sketch references (like points) or sketches itself.
4. **Pre-action** (Post-action): Add or move an existing action to a previous location in the system. CAD: Move operations before in the sequence.
5. **Inclusion** (Exclusion): To include patterns and sequences within a system. CAD: Patterns in sketches.
6. **Align** (Displace): Elements of a system are made homogeneous. CAD: Standardization of sketches or operations.
7. **Closed cycle** (Open cycle): A part is isolated from the system, having its own processes. CAD: A variable turns into a constant (outside the evolutionary cycle).

Looking at the proposed solutions for each alternative, several equivalences can be found:

- Fresnel Lens
  - 2.Minimize range (Maximize) points to the same spot than *universality*, in this case to unify the focal point regarding the incidence angle. However, it is considered that *universality* in the lens context is too general to give a real help and consequently is not clearly reflected in the results.
  - 5. Inclusion (exclusion) is a DiP able to generate viable alternatives, by including previous Fresnel patterns within a surface or volume. This could reproduce the idea of putting new receptors in critical spots with the capacity to refocus the received rays. The redirection of rays can also be interpreted as a manifestation of principle 4.Pre-action (Post-action).
- Minimum following tracking system
  - 1. Separation (unification) can be considered in the same way that *segmentation,* separating the components of the system that follows and concentrate the solar rays.

In a CAD context this should be complemented by 5.Inclusion (exclusion), in order to explore possible patterns in the mirror distribution.

- 7.Closed cycle (Open cycle) can define the needed periodic movement, complemented with 3.Balance (Unbalance) which suggests that the repositioning should be adequate to the solar movement, in order maximize the energy collected / energy consumed ratio.

Conceptually it is also possible to have a CAD based EA applying the DiP's by moving and reproducing mirrors and concentration spots under a fixed sun, situation that can be repeated for different daytimes. This could indicate the most efficient combinations to perform a second study under stricter conditions.

5.2.4. From the study case 4 (Section 4.5)

In this study case, the search of solutions started with an arbitrary idea tested in physical models (paper), and later simulated and optimized computationally. Since the author of the study is not completely satisfied whit the results (unwanted situation), the process is repeated utilizing a different concept, which obtains better results than the original one. About this, many speculations can be made:

- The background of the second designer (its intuition and expertise) allowed him to do a better design.
- The development of the first concept opened the mind of the designer, allowing him to improve further the component.
- The first concept was a poor one, easy to be surpassed.
- The second concept was a "lucky" one.

In a less speculative area, it was observed that the clear definition of restrictions and the simplicity of the analysis in early stages are critical to direct the path toward valid solutions. The importance of optimization must be remarked, since it can achieve the global objective with parameter adjustments, making unnecessary to look for a new Pareto front.

Similarly to the previous study cases, the approach to solve the problem was based in to complement the designers' intuition with the computer ability to simulate. First, an arbitrary idea is presented, then validated, and finally improved. This process is clearly within the DNA: the crash absorber is negated by the second concept, changing the search space (parameters location and potential values) constituting a new proposal to be optimized (see Figure 81). This is a level 3 modification, where the fundamental change was made by the designer.

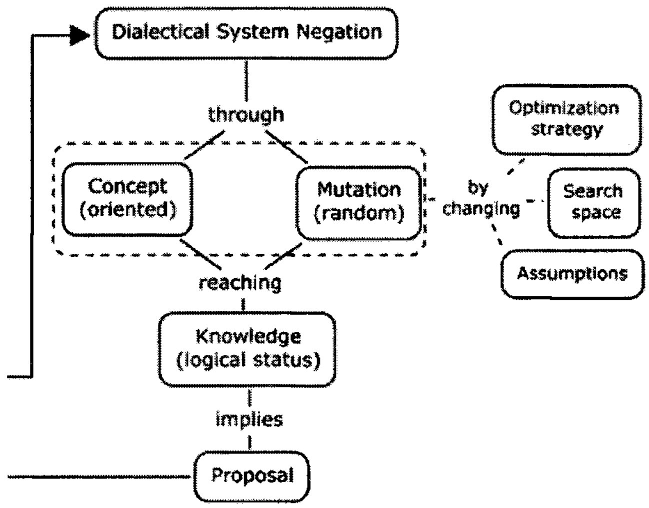

Figure 81: Part of the DNA, which describes the followed process during the concept generation.

About the concept generation, the first concept proposal could have been assisted by the DiP of *Inclusion* (includes patterns and sequences within a system) related to the two components: the bending patterns, and the idea of inserting one section into another. In the second case, *Minimize range* (affects the extension of the actions applied to a system) is the basis of the simplicity of the concept, and *Pre-action* (it affects the sequence of events performed by the system) is how the high force in the beginning is controlled by the circles diameters.

Finally, a relevant observation was made during the study of the case: despite there was still an "unwanted situation" (human defined, in this case ambiguously), the computational objective didn't change. This is interpreted as that within the DNA, whenever using computers, the **unwanted situation** should be a specific **objective** that hasn't been reached, being a first step to align those two cycles: the human and the computational one.

5.2.5. General reflections

In order to aid the different stages of a computational product development process, the basic structure of any tool intended for human use has to be first understood, and then modified. In the presented study cases, EAs and TRIZ are integrated using Dialectics:

- EAs are adapted to be part of the DNA cycle
- TRIZ 40-IP are reformulated into the DiP to be utilized inside the DNA cycle
- It is expected that this integration can lead to the solution of inventive problems situations in a more efficient manner.

Inventing requires looking in different directions (not necessarily new-unknown ones) so the inclusion of another "search axis" is a necessary step once the optimization has been performed, implying also the need of defining the axis. From this point of view, what establishes the difference between optimization and invention performed by computers is the capability of automatically expanding the search space for enhanced performance.

However, **the criteria to adding a new search axis is the main challenge** of this approach because theoretically there are infinite options, and the exploration of each one would increase exponentially the computational time. The study cases suggests that if no direction of solution is known, through preliminary evaluations the most promising DiP for the particular problem can be analyzed based on performance, with almost no human intervention (and thus prejudgment and mental inertia, among others). However, if there is available information about how to solve the problem, the human intervention could be made from the moment of selection of the operators, which would save considerable amounts of processing time. In other words, the problem characteristics and the user's level of expertise, from novice to visionary (Eder 2009), will determine the adequate operators and computational force to be applied.

The idea of dealing indirectly with the search space reduction and solution's quality measurement, ("killing the weak", section 4.2.1) proved to be conceptually viable. To explore this in real world problems, like the solar concentrators, it is necessary to go deeper in how to define the parameters, hierarchies and restrictions to be edited by the algorithm. The "softer" alternative that the DiP guides the user in the space expansion-modification is already being developed commercially in software like the previously mentioned IWB and Goldfire innovator.

Finally, a problem cannot be defined computationally since is a human perception. The objective is what needs to be defined to aid the problem solving, meaning that a DNA "unwanted situation" computationally is an objective that hasn't been reached yet. **It is not problem solving, but objective achievement.** Each cycle, human or computational, can follow its own path.

5.3.Latest developments

During the analysis of the study cases several ideas regarding the completeness and format of the DNA and the DiP gained force.

5.3.1. DNA modifications

DNA is not declaring two stages that seem to be fundamental in problem solving:

- Codification: Systematic arrangement of data in order to efficiently re-utilize it.
- Assimilation: to incorporate the codified data into the IPS process.

To understand this, it can be observed how case 2 was solved using the codified data of commercial alternatives, in this case probably in engineers heads. In case 4, the assimilation of the previous developed shapes is the most probable explanation for the improved performance obtained at the first new attempt. It can be said that they are essential directing evolution, becoming the "shoulders of giants".

Another point that is not explicitly mentioned is the importance of a clear objective, especially when CS is going to be used: problems are a human perception, and computers have no way to deal with something that is defined ambiguously.

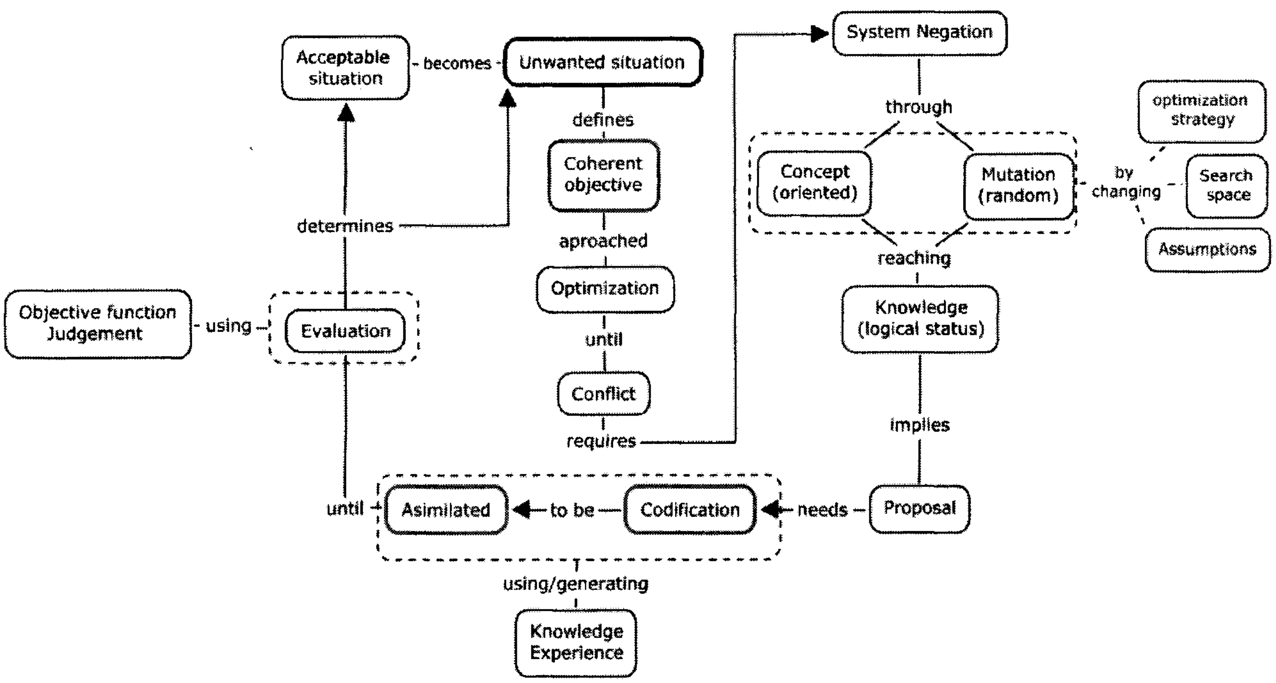

Figure 82: New DNA, including three new stages. First, after an unwanted situation is declared, it is necessary to define a concrete objective, measurable and consistent. Also, after any proposal the generated knowledge must be codified and assimilated, in order have a repeatable and reliable as possible procedure.

5.3.2. DiP modifications

It was found that "revert" an action or a relation was a useful principle, sometimes confused with negating the situation and sometimes considered too indirectly.

An easy way to distinguish both cases is that to revert something implies to maintain the system, and to negate demands to do something different, beyond magnitudes. For example, both solar collectors inverted physically the concentration focus position. Revert's opposite is to do more of the same, to "reinforce", also a common solution.

Another observation is that there exist an implicit distinction between time and space when applying the DiP. This is considered artificial, since together they constitute the sensible reality, so they must be part of every principle. Following the same line of thought, some of the principles are oriented to the components of a system and other to its relations, which is another unnecessary distinction since both are just system components.

All the previous suggest that the DiP should be restructured, process that is resumed in Figure 83 and exemplified in Table 12.

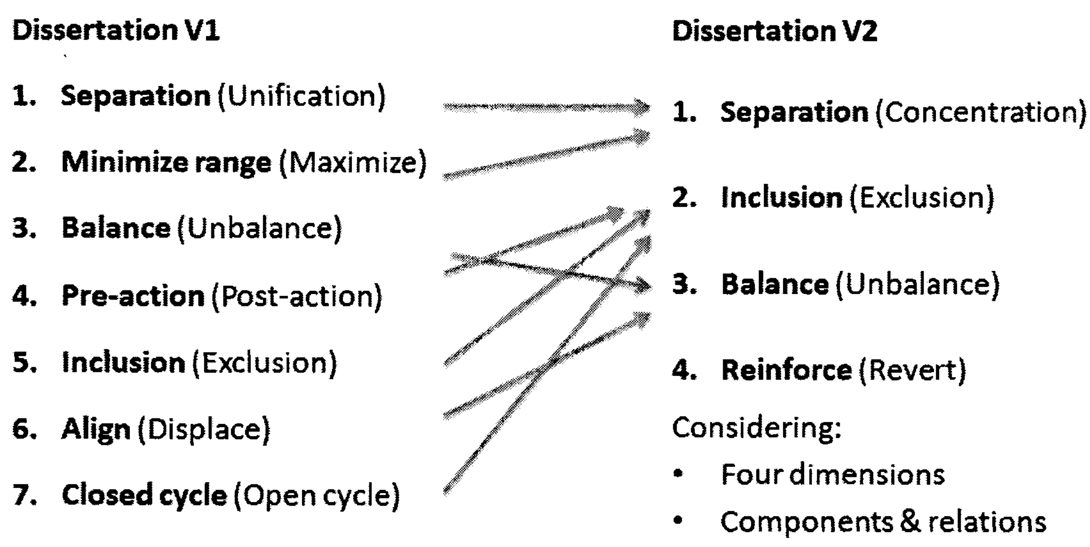

Figure 83: Equivalences between the 7DiP already presented and the new 4 ones, which consider together with the duality, the application to components, relations, time, and space.

Table 12: Definitions and examples of the DiP in their new format.

"New" Principles	Applied to system's **components** or **relations**, considering the **4 Dimensions**	Examples from the 40-IP
1.Separation (Concentration)	A component of the system, or its participation, is divided-extended.	Component: Segmentation Relations: Merging Both: Skipping
2.Inclusion (Exclusion)	New elements or relations are included into the system.	Component: Mechanics substitution Relations: Inert atmosphere Both: Preliminary action
3.Balance (Unbalance)	Component or relations are modified according to a distribution.	Component: symmetry Relations: Equipotentiality Both: Homogeneity
4.Reinforce (Revert)	A component or the way how it relates to others is reinforced	Component: Phase transitions Relations: The other way round Both: Partial or excessive actions

Several hybrid examples can be mentioned too: Intermediary (1 y 2), another dimension (1 y 3), local quality (1 y 4), anti-weight (2 y 3), preliminary anti-action (2 y 4), continuity of useful action (3 y 4), and others. The essential of the new structure is presented in Figure 84.

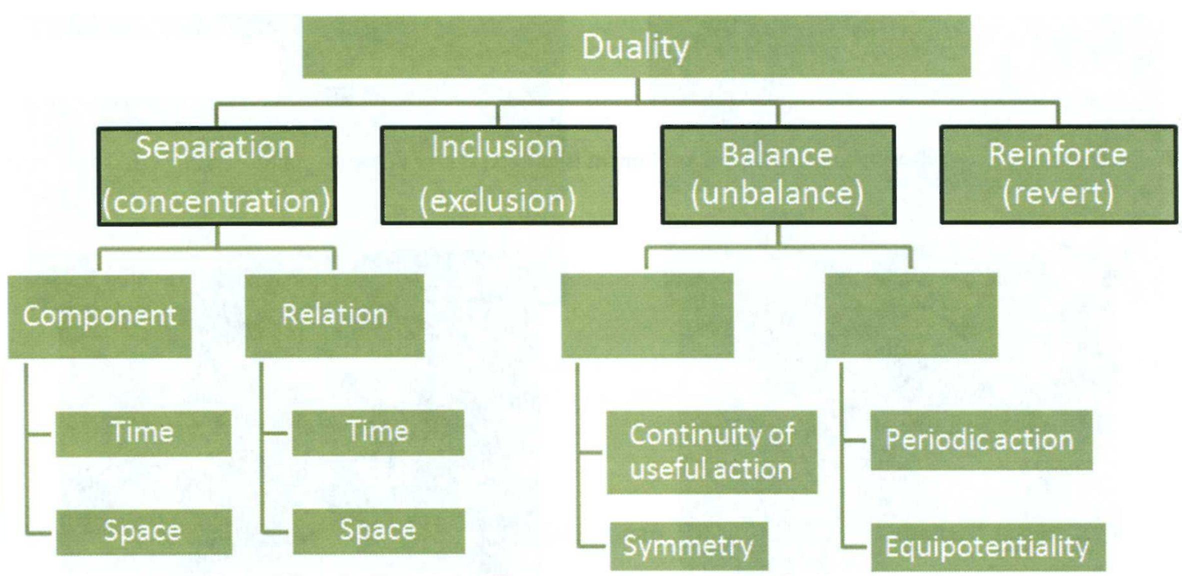

Figure 84: Structure to construct and mix principles from the 4 basic dialectical ones. The direct quantity (without mixing) is 4 principles * 2 (duality) * 2 (dimension) * 2 (system) = 32 principles. Four branches of Balance are exemplified using some of the 40-IP.

5.3.3. Example of DiP's last structure: evolution of a solar energy storage tank

The following is a "phase change material energy storage device", being developed at the present (Cárdenas et al. 2011). Here is briefly analyzed its evolution and potential development considering the DiP as a framework.

The tank concept is intended to work as an energy reserve for a domestic tri-generation system utilizing a Stirling engine to produce electric power. The thermal energy storage saves a fused material and its latent energy inside the tank, obtaining this way a great energy density easy to be extracted with very small variations on the temperature. Each of the following figures presents a modification and how it can be "fitted" within the DiP.

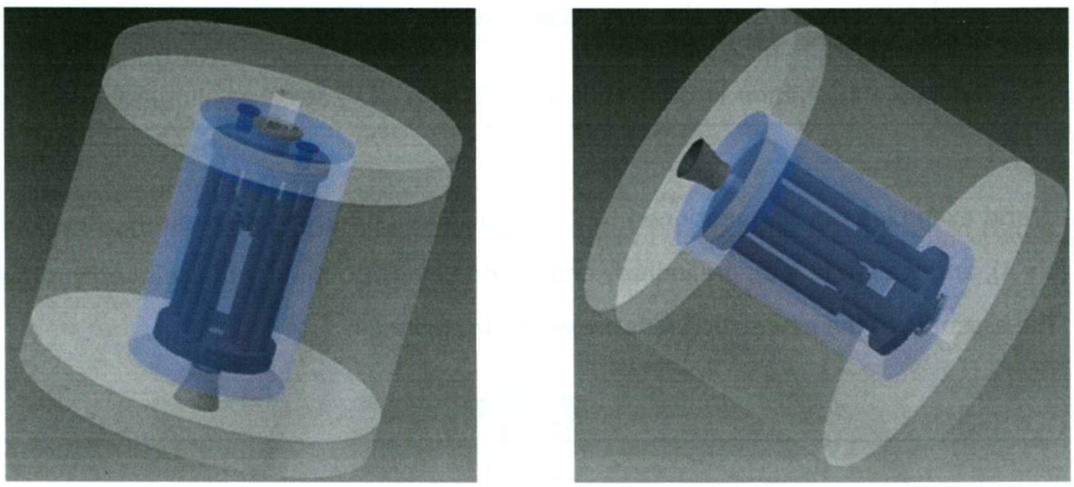

Figure 85: The inferior exit is removed: Exclusion, an element is excluded from system (applied to a component, in space)

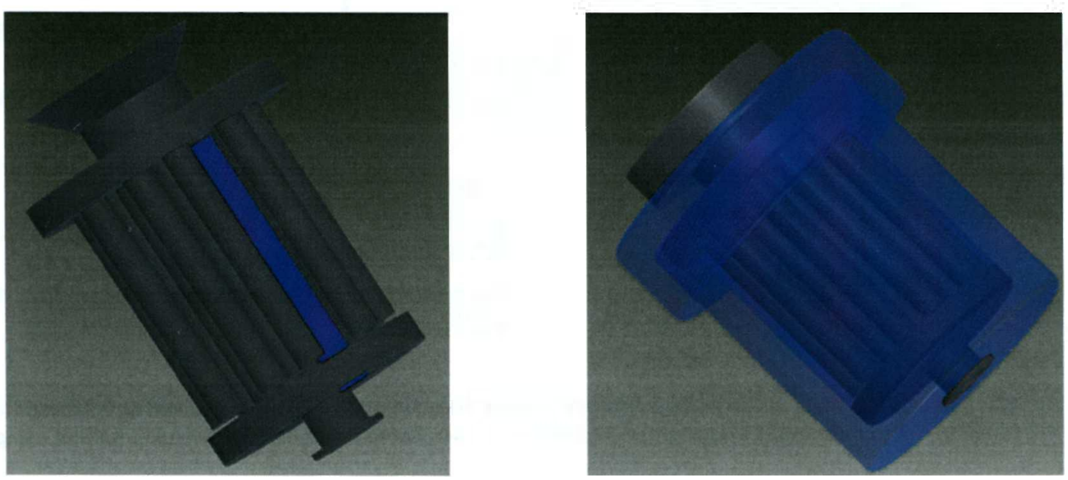

Figure 86: The conductive bars are distributed in a uniform pattern: Balance, a modification according to a distribution (a relation, in space)

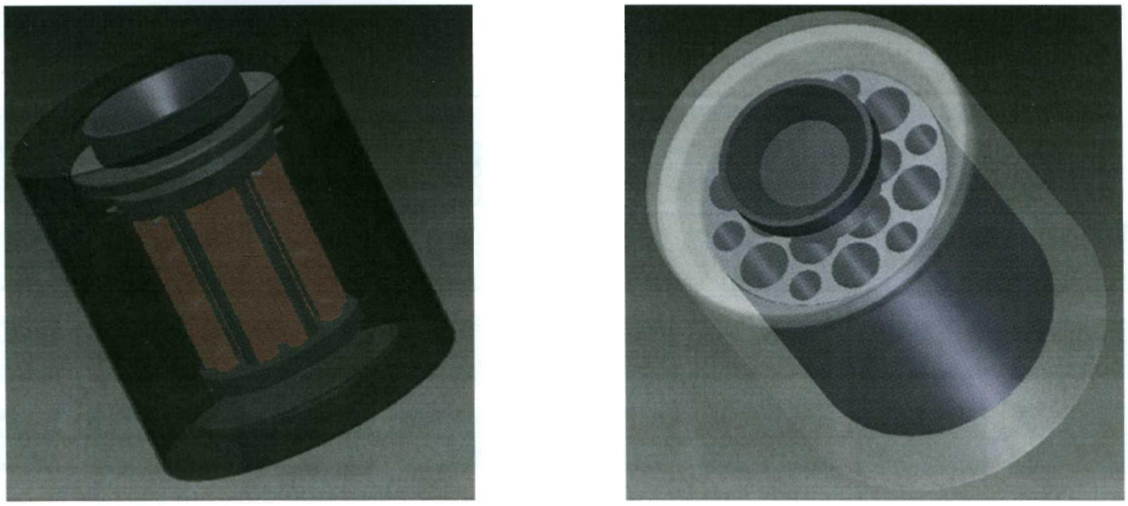

Figure 87: The space occupied by the bars is reverted: Revert (in space)

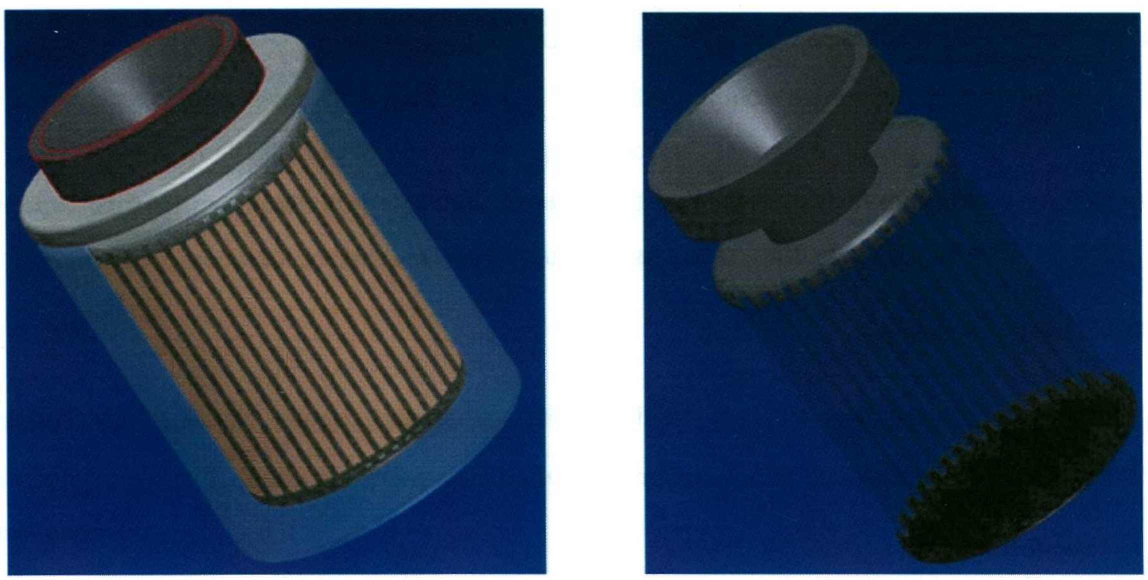

Figure 88: The space occupied by the conductor is distributed along the body: Separation and Balance. Here the components and its relations are spatially edited.

Looking at the structure presented in Figure 84, it can be seen that several principles hasn't been explored, especially when related with their application in different time frameworks.

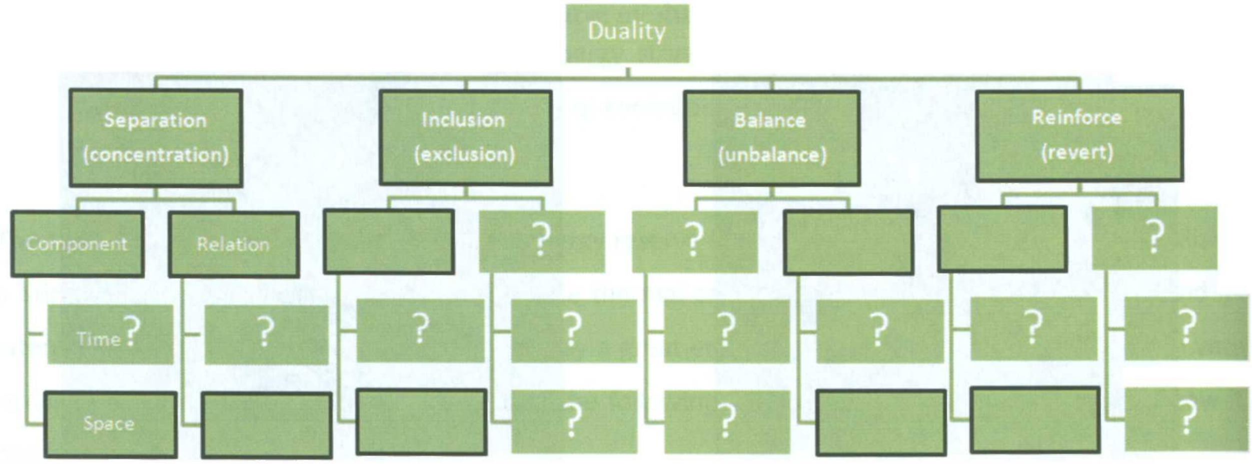

Figure 89: DiP basic structure diagram, showing the principles developed so far (without question sign).

From the several alternatives to explore, some ideas can be suggested:

- Inclusion/exclusion

Relation: to include a material that changes it function in time (e.g. with high thermal expansion), that cuts the flow with low temperatures and reconnects it with high ones.

- Balance/unbalance

Component: analyze the effect of modifying the component regular shape of the conductive

- Reinforce/revert

Relation: the space utilized by the storage material can be changed in time, i.e. the melt liquid "leaves" the conductor to be isolated.

The next stage to consider is analyze the DiP's mixed, under more stricter rules (restrictions).

6. Research conclusions

6.1.Summary

The macro-problem is *problem solving*, but why attempt to do it using CS, why is valuable? The main reason is, as shown in the literature review, the dependence on *intuition* and *expertise*. This increases the design process unpredictability and cost.

The three questions presented in the introduction (section 1.2) are the guide to present the most relevant conclusions:

1. Which are the problems when solving problems?
2. How actual methods are dealing with those problems, including the user?
3. How to make a model as user independent as possible, increasing the automation of IPS in order to utilize CS to actively aid the concept generation?

A relevant observation has to be made: several areas that supposedly where considered in literature where not treated at all (like section 3.3), or trapped in loops[20] (like section 2.1.8). This lack of studies to compare with has several effects, which can be resumed in the absence of references especially during the analysis. It is expected that the previous has not biased the conclusions.

6.2.Conclusions and implications

Regarding question 1, there are many situations to consider during problem solving, being **the most critical ones**:

- Mental inertia
- Meaningful Learning
- Analysis of the actual situation
- Assistance in the creation of solutions
- Evaluation of the concept

This list was obtained from the construction and analysis of several networks of problems (section 3.2.2). A method that does not treat them is considered, at most, a tool that can be helpful in specific

[20] A good example is the morphological matrix, proposed as a computational alternative to find concepts in relatively recent articles, like (Strawbridge et al. 2002)

situations. At this level of analysis, *Meaningful learning* was the less treated in the IPS literature, considered usually a given.

In this list the user is critical, existing a clear conflict between having a deep & useful knowledge and to obtain it quickly & effortless. The conclusion is that, in order to solve inventive problems effectively, there must be a ***balance* between the user level of expertise and the degree of detail of the method utilized** (section 3.4.1), being one extreme the total dependence of the user heuristic (intuition) and the opposite the total dependence on the procedure (automation).

Continuing with question 2, the literature tends to depends on the user, becoming more important than a specific tool the **learning and application of the method selected to confront an undesired situation**. This can be seen as an example of the relevance of *mental inertia* along the IPS, and its complementary need of *meaningful learning* (specially the will to learn). In general, as shown in Table 7, the critical problems are not being considered by the different IPS methods, relying solely on heuristics for the generation of concepts.

Question 3 is the central reason of the present research, and is composed of several parts. The model proposed (section 3.5.2) is based on the negation that a system must go through in order to cross the optimization barrier (the Pareto front) being fundamental since it **defines the conflict that demands concepts**. This also explains why optimization must precede negation.

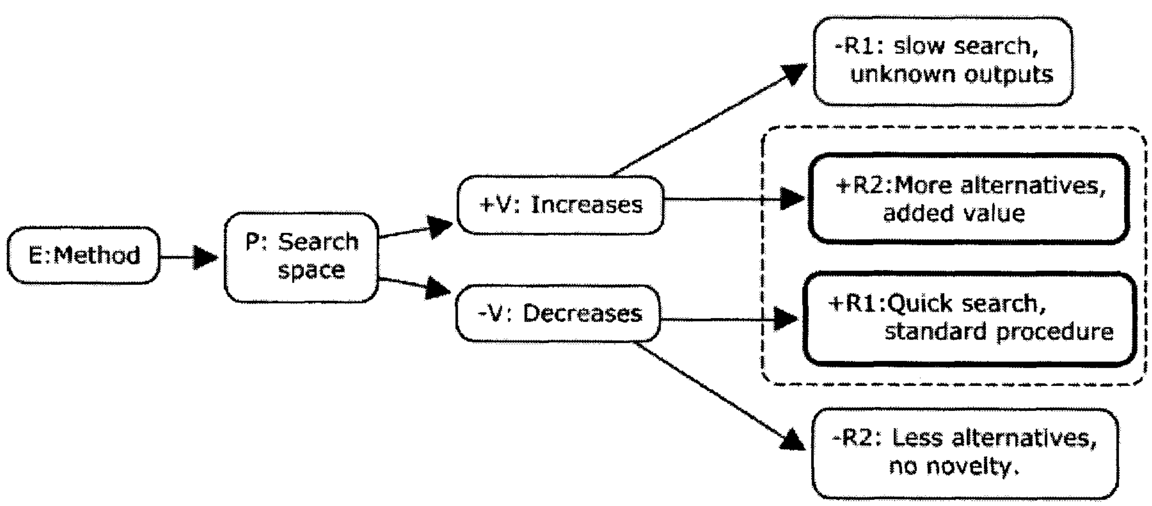

Figure 90: A clear conflict appears when a system is negated, originated on the implied expansion of the search space. The utilization of the DiP and CS are the intended way to approach the ideal final result.

The DiP are a concrete improvement when looking for guidance (not inspiration condensing a great amount of applicable knowledge, modeling better than the state of the art the resolution of problems.

With the model and the principles clear, CS can be explored about to aid concept generation. The DiP were able to be integrated to a GA, delivering emerging information (user deducted) and opening the possibilities of influencing the output in a more predictable manner.

The *inverted selection* showed to be a valid alternative to attack the difficulty of evaluation, discarding the bad elements (as science does with unsatisfactory theories) instead of selecting the good ones. This seems to be a better approach when no exact selection criteria for the better solutions are available, as often occurs in inventive processes: developers do not know specifically what they are looking for, but they are clear about what is not an acceptable answer. Regarding the extra time needed to reach the Pareto front, this is a clear disadvantage for optimization problems, but this dissertation does not aim to optimize.

In general, in order to effectively computationally assist the concept generation, it is necessary to define not only the involved parameters in the model, but the potential relations, hierarchies and restrictions between each other. **The restrictions and the objective function are the ones that define a problematic** and to modify them implies destroying the system, consequently, they define the limits of the computational assistance. Once they are clarified, the *dialectical system negation* can start being automated. If a system does not have the ability of editing its components, it cannot cross the Pareto barrier.

6.3.Future research needs

The contribution of this dissertation is fundamentally conceptual, mainly because until now the study of the topic has being random and with no clear distinction from optimization, and in many times under the belief that creativity is unknowable. In order to obtain practical results, several stages need to be accomplished:

- To analyze a set of standard problems in order to compare the performance of traditional evolutionary operators v/s dialectical ones, improving them if necessary.
- Determine the situations on which each DiP is more adequate, complemented for example by a design of experiments.

After this first stage, is necessary to define the *dialectical negation* criteria:

- Determine the computational problems that the integration of new axis creates.
- Define the procedure to edit parameters: values, relations between, restrictions.
- Determine the most adequate "language" to use (GAs, GP, shape grammars, topological...).
- Validate the previous by finding alternative solutions (or replicating) to solved problems, comparing also the performance of traditional evolutionary operators v/s DiP.

There are many ideas to explore, like editing the *parameters* indirectly: in the initial formulation of the problematic an extra number of parameters can be included, by this, the objective is reversed towards discarding search axes instead of adding new ones, a process that can be guided by the DiP based on historical performance. However without dealing with the previously mentioned stages, those ideas will never become concepts able to computationally aid problem solving.

In a non computational context, the DNA-DiP potential can be measured in students, comparing the performance of different groups under standard conditions. This will also determine the need of training in situations where the expertise levels are variable.

Bibliography

Albano, L.D. & Suh, N.P., 1992. Axiomatic approach to structural design. *Research in Engineering Design*, 4(3), pp.171-183.

Albers, A. et al., 2009. Development of an engine crankshaft in a framework of computer-aided innovation. *Computers in Industry*, 60(8), p.604–612.

Altshuller, G., 1984. *Creativity As an Exact Science* 1st ed., Gordon and Breach Science Publishers.

Altshuller, G., 1999. *The Innovation Algorithm: TRIZ, systematic innovation and technical creativity*, Technical Innovation Ctr.

Altshuller, G. & Shulyak, L., 1998. 40 Principles: TRIZ Keys to Technical Innovation. *Translated by Lev Shulyak, Technical Innovation Center, Worcester, MA*.

Antony, J., 2003. *Design of Experiments for Engineers and Scientists* 1st ed., Butterworth-Heinemann.

Barry, K., Domb, E. & Slocum, M.S., 2008. TRIZ-What Is TRIZ. *The TRIZ journal Content*.

Bentley, P.J. & Wakefield, J.P., 1997. Conceptual evolutionary design by a genetic algorithm. *Engineering design and automation*, 3, p.119–132.

Bentley, P.J. ed., 1999. *Evolutionary Design by Computers*, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

Bouchard, T.J., 1972. A comparison of two group brainstorming procedures. *Journal of Applied Psychology*, 56(5), p.418–421.

Brostow, A., 2006. Implementing triz at air products. *Knowledge Management Review*, 9(3), p.26.

Cárdenas, B., Leon, N. & Garcia, L., 2011. Tanque para almacenamiento de energía solar térmica. In IX Congreso Internacional sobre Innovación y Desarrollo Tecnológico. Cuernavaca, Mexico.

Chong, Y.T., Chen, C.-H. & Leong, K.F., 2008. A heuristic-based approach to conceptual design. *Research in Engineering Design*, 20(2), pp.97-116.

Coello, C.A., 2000. An updated survey of GA-based multiobjective optimization techniques. *ACM Computing Surveys (CSUR)*, 32(2), pp.109-143.

Cueva, J.M., 2006. *Automatic shape variations for optimization purposes*. Monterrey: Instituto de Estudios superiores Tecnologico de Monterrey.

Deb, K. et al., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2), pp.182-197.

Deb, K. & Srinivasan, A., 2008. INNOVIZATION: Discovery of Innovative Design Principles Through Multiobjective Evolutionary Optimization. *Multiobjective Problem Solving from Nature*, p.243–262.

Deb, K. & Srinivasan, A., 2006. Innovization: Innovating design principles through optimization. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. p. 1636.

Dieter, G., 1999. *Engineering Design: A Materials and Processing Approach* 3rd ed., McGraw-Hill Science/Engineering/Math. Available at: http://www.mhhe.com/engcs/mech/dieter/.

Dieter, G. & Schmidt, L., 2009. *Engineering Design*, McGraw-Hill.

Dung, P., 1997. Dialectical Systems Thinking (TRIZ) (Phan Dung). Available at: http://www.osaka-gu.ac.jp/php/nakagawa/TRIZ/eTRIZ/epapers/ePhanDungP990708/ePhanDungPA990708.html [Accessed October 17, 2008].

Duran-Novoa, R., Leon-Rovira, Noel & Uresti, E., 2009. A Dialectical Approach to Solve Inventive Problems Using Genetic Algorithms and TRIZ: Searching For a Computer Aided Innovation Shell. In *Proceedings of the 17th International Conference on Engineering Design (ICED'09)*. International Conference on Engineering Design, ICED'09. Stanford University, Stanford, CA, USA: The Design Society, pp. 415-426.

Eder, W.E., 2009. Design Engineering and Needs for Methodology. In *Proceedings of the 17th International Conference on Engineering Design (ICED'09)*. International Conference on Engineering Design, ICED'09. Stanford University, Stanford, CA, USA: The Design Society, pp. 1-12.

Engels, F., Dutt, C. & Haldane, J.B.S., 1964. *Dialectics of Nature*, Progress Publishers.

Goldberg, D. E, 1991. Genetic algorithms as a computational theory of conceptual design. In *Applications of Artificial Intelligence in Engineering, Proceedings of the 6th International Conference on Artificial Intelligence in Engineering (AIENG91)*. p. 3–16.

Goldberg, David E., 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning* 1st ed., Addison-Wesley Professional.

Gordon, W.J.J., 1961. *Synectics: The development of creative capacity*, Harper & Row New York.

Hatchuel, A. & Weil, B., 2003. A new approach of innovative design: an introduction to CK theory. In *Proceedings, International Conference on Engineering Design*.

Hatchuel, A. & Weil, B., 2009. CK design theory: an advanced formulation. *Research in Engineering Design*, 19(4), p.181–192.

Hatchuel, A. & Weil, B., 2002. CK Theory: Notions and applications of a unified design theory. In *Proceedings of the Herbert Simon International Conference on" Design Sciences", Lyon*. p. 22.

Helander, M.G. & Lin, L., 2000. Anthropometric design of workstations. In *Proceedings of the first International Conference on Axiomatic Design*. p. 130–138.

Horowitz, R. & Maimon, O., 1997. Creative design methodology and the SIT method. In *ASME Design Engineering Technical Conference*. p. 14–17.

Jiao, J., Zhang, Y. & Wang, Y., 2007. A generic genetic algorithm for product family design. *Journal of Intelligent Manufacturing*, 18(2), p.233.

Khomenko, N. & De Guio, R., 2007. OTSM Network of Problems for representing and analysing problem situations with computer support. *Trends in Computer Aided Innovation*, p.77–88.

Khurana, A. & Rosenthal, S.R., 1997. Integrating the fuzzy front end of new product development. *Sloan Management Review*, 38, p.103–120.

Kicinger, R., Arciszewski, T. & Jong, K.D., 2005. Evolutionary computation and structural design: A survey of the state-of-the-art. *Computers and Structures*, 83(23-24), pp.1943-1978.

Koza, J.R., Streeter, M.J. & Keane, M.A., 2008. Routine high-return human-competitive automated problem-solving by means of genetic programming. *Information Sciences*, 178(23), pp.4434-4452.

Koza, J. et al., 2004. Invention and creativity in automated design by means of genetic programming. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing : AI EDAM*, 18(3), p.245.

Kucharavy, D., 2006. Module 5: TRIZ methods and tools (http://www2.insa-strasbourg.fr/triz/amid.php). In *Handouts of the Advanced Master in Innovative Design*.

K-Z Chen and X-A Feng, 2004. Virtual genes of manufacturing products and their reforms for product innovative design. *Proceedings of the Institution of Mechanical Engineers*, 218(5), p.557.

Lee, H. & Tang, M., 2009. Evolving product form designs using parametric shape grammars integrated with genetic programming. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing : AI EDAM*, 23(2), p.131.

Lewis, J., Engels, Friedrich & Marx, K., 1932. *Introducción al Estudio de la Filosofía y del Materialismo Dialéctico (Versión de M.Ortuzar)* First edition., Ediciones Frente Cultural México. D. F.

Malmqvist, J., Axelsson, R. & Johansson, M., 1996. A comparative analysis of the theory of inventive problem solving and the systematic approach of pahl and beitz. In *The 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*. Irvine, California.

Mann, D., 2004. Comparing The Classical and New Contradiction Matrix Part 1- Zooming Out. *The Triz Journal*. Available at: http://www.triz-journal.com/archives/2004/04/01.pdf [Accessed November 11, 2008].

Mann, D., 2002. Evolving the Inventive Principles. *The Triz Journal*. Available at: http://www.triz-journal.com/archives/2002/08/d/index.htm [Accessed October 27, 2008].

Mann, D. et al., 2003. *Matrix 2003 Updating the TRIZ contradiction Matrix* 1st ed., Belgium: CREAX Press.

Martinez, O., 1999. *Integración de metodologías de diseño innovativo y de optimización paramétrica aplicadas en el diseño de una viga de frenado de ferrocarril*. Instituto de Estudios superiores Tecnologico de Monterrey.

McPhee, N.F., Ohs, B. & Hutchison, T., 2008. Semantic building blocks in genetic programming. In *Proceedings of the 11th European conference on Genetic programming*. p. 134–145.

Messac, A. & Mattson, C.A., 2004. Normal constraint method with guarantee of even representation of complete pareto frontier. *AIAA journal*, 42(10), p.2101–2111.

Nakagawa, T., 2006. A New Paradigm for Creative Problem Solving: Six-Box Scheme in USIT. In *ETRIA TFC2006, at Kortrijk, Belgium, on October*. p. 6–8.

Nakagawa, T., 2000. USIT–Creative Problem Solving Procedure with Simplified TRIZ. *Journal of Japan Society for Design Engineering*, 35(4), p.111–118.

Novak, J.D. & Cañas, A.J., 2008. The theory underlying concept maps and how to construct and use them. *Florida Institute for Human and Machine Cognition Pensacola Fl. http://cmap.ihmc.us/Publications/ResearchPapers/TheoryCmaps/TheoryUnderlyingConceptMaps.htm*.

O'Reilly, U.M. & Oppacher, F., 1995. The troubling aspects of a building block hypothesis for genetic programming. *Foundations of Genetic Algorithms*, 3, p.73–88.

Osborn, A., 1963. *Applied Imagination; Principles and Procedures of Creative Problem-Solving* 3rd ed., Scribner.

Otto, K.N. & Wood, K.L., 2001. *Product Design: Techniques in Reverse Engineering and New Product Development*, Prentice Hall.

Pahl, G. et al., 1996. *Engineering Design: A Systematic Approach*, Springer Verlag.

Pearl, J., 1984. Heuristics: intelligent search strategies for computer problem solving.

Platt, J.R., 1964. Strong inference. *science*, 146(3642), p.347–353.

Poli, R., Langdon, W.B. & McPhee, N.F., 2008. *A field guide to genetic programming*, Lulu Enterprises Uk Ltd.

Polya, G., 1988. *How to Solve It: A New Aspect of Mathematical Method*, Princeton University Press.

Renner, G. & Ekart, A., 2003. Genetic algorithms in computer aided design. *Computer-Aided Design*, 35(8), p.709–726.

Ritchey, T., 1998. Fritz Zwicky Morphologie and Policy Analysis. In *16th Euro Conference on Operational Analysis, Brussels*.

Robertson, B.F. & Radcliffe, D.F., 2009. Impact of CAD tools on creative problem solving in engineering design. *Computer-Aided Design*, 41(3), pp.136-146.

Serway, R.A. et al., 2000. *Physics for scientists and engineers*, Saunders College Publishing.

Sewe, E., 2008. *Structural optimization of a crash absorption component*. Hamburg University of Applied Sciences.

Silverstein, D., DeCarlo, N. & Slocum, M., 2007. *Insourcing Innovation: How to Achieve Competitive Excellence Using TRIZ*, Auerbach Publications.

Simon, H.A., 1996. *The sciences of the artificial*, The MIT Press.

Smyth, M. & Edmonds, E., 2000. Supporting design through the strategic use of shape grammars. *Knowledge-Based Systems*, 13(6), pp.385-393.

Strawbridge, Z., McAdams, D.A. & Stone, R.B., 2002. A computational approach to conceptual design. In *Proceedings of DETC2002, DETC2002/DTM-34001, Montreal, Canada.*

Suh, N.P., 1998. Axiomatic Design Theory for Systems. *Research in Engineering Design*, 10(4), pp.189-209.

Tversky, A. & Kahneman, D., 1974. Judgment under uncertainty: Heuristics and biases. *Science, New Series*, 185(4157), pp.1124-1131.

Ulrich, K. & Eppinger, S., 2003. *Product Design and Development* 3rd ed., McGraw-Hill/Irwin.

Valenzuela-Rendon, M., 2003. The Virtual Gene Genetic Algorithm. *Lecture Notes in Computer Science*, pp.1457-1468.

Vidal, R., 2009. Creativity for problem solvers. *AI & Society*, 23(3), p.409.

Wen Luo and Chaoan Lai, 2009. A new product design method based on integrated intelligence. In *Computer-Aided Industrial Design & Conceptual Design, 2009. CAID & CD 2009. IEEE 10th International Conference on*. Computer-Aided Industrial Design & Conceptual Design, 2009. CAID & CD 2009. IEEE 10th International Conference on. pp. 638-642.

Williams, S., 2005. Unnatural Selection. *Technology Review*, 108(2), p.54.

Yajot, O., 1973. *Que es el Materialismo Dialéctico*, Moscú: Editorial Progreso.

List of Figures

List of Tables

Glossary

- **Assimilation**: to incorporate the codified data into the IPS process.
- **Automatic**: Device, procedure, or system that self-activates (or is executed) under programmed or specified conditions and performs specified functions. **Automaticity**: Fluent performance without the conscious deployment of attention. It is the ability to do things without occupying the mind with the low level details required, being usually the result of learning, repetition, and practice.
- **Capacity**: Specific ability of an entity (person or organization) or resource, measured in quantity and level of quality, over an extended period.
- **Concept**: clear, detailed description of the attributes and benefits of a new product that addresses the needs of certain customer. The reasoning behind an idea, strategy, or proposal with particular emphasis placed on the benefits brought on by that idea. **Design concept**: an idea that is sufficiently developed that it can be evaluated in terms of physical realizability; i.e., the means of performing each major function has been determined.
- **Conflict**: incompatibility (real or perceived) between opinions, principles, characteristics, etc, of a system. In technical systems, for example, happens when to improve further one parameter affects negatively another.
- **Creativity**: Mental characteristic that allows a person to think outside of the box, which results in innovative or different approaches to a particular task.
- **Computer:** Programmable machine that receives input, stores and manipulates data, and provides output in a useful format
- **Computer science (CS)**: is the study of the theoretical foundations of information & computation, and of practical techniques for their implementation and application in computer systems. It is frequently described as the systematic study of algorithmic processes that create, describe, and transform information. According to Peter J. Denning, the fundamental question underlying computer science is, "What can be (efficiently) automated?" (Information: ordered sequence of symbols; Computation: Any type of information processing.)
- **Cybernetics**: Is the study of feedback and derived concepts such as communication and control in living organisms, machines and organizations. Its focus is how anything (digital, mechanical or biological) processes information, reacts to information and changes or can be changed to better accomplish the first two tasks.
- **Codification**: Systematic arrangement of data in order to efficiently re-utilize it.

- **Data**: Description of selected events from a certain viewpoint (the correspondent information and knowledge one) using known parameters and values (quantitative and qualitative) to measure the event in relation to something known (size, color; strength).
- **Design concept**: an idea that is sufficiently developed that it can be evaluated in terms of physical realizability; i.e., the means of performing each major function has been determined.
- **Designer**: person who devises courses of action aimed at changing existing situations into preferred ones. It is the principal mark that distinguishes the professions from the sciences.
- **Dissertation**: is a lengthy, formal document that argues in defense of a particular **thesis** (a hypothesis or conjecture). The research performed must be both, original and substantial.
- **Expertise:** Capacity to immediately recognize the type and requirements of an analyzed situation. Its abnormalities, strengths, weaknesses and available resources (e.g. data, personnel) are considered, especially what is missing and what is useless.
- **Expertise levels**:
    - **Novice**: A novice will consider the objective features of a situation, as they are given by the experts, and will follow strict rules to deal with the problem.
    - **Advanced beginner**: For an advanced beginner the situational aspects are important, there is sensitivity to exceptions to the 'hard' rules of the novice. Maxims and heuristics are used for guidance through the problem situation.
    - **Competent**: A competent problem solver selects the elements in a situation that are relevant, and chooses a plan to achieve the goals. This selection and choice can only be made on the basis of a much higher involvement in the design situation than displayed by a novice or an advanced beginner. Problem solving at this level involves the seeking of opportunities, and of building up expectations. At this level of involvement the problem solving process takes on a trial-and-error character, and there is a clear need for learning and reflection, that was absent in the novice and the beginner.
    - **Proficient**: A problem solver that then moves on to be proficient immediately sees the most important issues and appropriate plan, and then reasons out what to do.
    - **Expert**: The real expert responds to specific situation intuitively, and performs the appropriate action, straightaway. There is no obvious problem solving and reasoning that can be distinguished at this level of working. This is actually a very comfortable level to be functioning on, and a lot of professionals do not progress beyond this point – most do not need to.
    - **Master**: With the next level, the master, a new uneasiness creeps in. The master sees the standard ways of working that experienced professionals use not as natural but as contingent. A master displays a deeper involvement into the professional field as a whole, dwelling on success and failures. This attitude requires *an acute sense of context*, and openness to subtle cues. In his/her own work the master will perform more nuanced appropriate actions than the expert.
    - **Visionary**: The world discloser or 'visionary' consciously strives to extend the domain in which he/she works. The world discloser develops new ways things could be, defines the issues, opens new worlds and creates new domains. To do this a world discloser operates more on the margins of a domain, paying attention to other domains as well, and to anomalies and marginal practices that hold promises for a new vision of the domain.
- **Fuzzy Front End**: is the messy "getting started" period of new product development processes, where the organization formulates a concept of the product to be developed and decides whether

or not to invest resources in the further development of an idea. It ends when an organization approves and begins formal development of the concept.

- **Guidelines**: Recommended practice that allows some discretion or leeway in its interpretation, implementation, or use.
- **Heuristics**: strategies using readily accessible, though loosely applicable, information to control problem solving in human beings and machines.
- **Ideality (of a system)**: is the sum of benefits that provides divided by the sum of all costs and harms. A more "ideal system" should have all the benefits of the current one, other benefits, none of the limitations, and should not include any new limitations or difficulties.
- **Information:** articulated knowledge that through cognitive effort has become focal and structured. Structured interpretation (from a certain viewpoint) that giving meaning to events turns them into data.
- **Integration**: Process of attaining close and seamless coordination between several departments, groups, organizations, systems, etc. **Integrate**: To join together or unify disparate components or elements into a coordinated or harmonious whole.
- **Intuition**: Unconscious thought process that produces rapid, uninferred knowledge or solution. Though it is not analytic in the sense that it does not deliberately look for cause-and-effect (causal) relationships, intuition is not mere guesswork. Instead, it draws on previously acquired experiences and information and directly apprehends a totality. Intuition can be visionary or delusionary, uncannily correct or horrendously wrong in its conclusions.
- **Invention**: Something novel (within the context) and useful. Usually it is considered to be the result of creative thought.
- **Inventive problem**: human perception of a situation that has to be changed, but with at least one obstacle (or contradiction) that needs to be resolved, to achieve the desired goal.
- **Knowledge**: personal (tacit) and/or socially accepted (explicit) way of develop and using information and data to manage practical or intellectual tasks.
- **Linkages**: Relationships and interactions between tasks, functions, departments, and organizations, that promote flow of information, ideas, and integration in achievement of shared objectives.
- **Metaphor**: Method of extending the scope of a word, or of illustrating a point by showing agreement or correspondence (analogy) in details between two very different items. Unlike similes, metaphors do not employ the words 'like' or 'as' but instead make a direct reference without any explicit hint of comparison: cash cow, escalator clause, high flyer, lame duck, spinoff, etc.

- **Method**: Established, habitual, logical, or prescribed practice or systematic process of achieving certain ends with accuracy and efficiency, usually in an ordered sequence of fixed steps (e.g. Scientific Method).
- **Methodology**: System of broad principles or rules from which specific methods or procedures may be derived to understand different situations (or solve different problems) within the scope of a particular discipline. Unlike an algorithm, a methodology is not a formula but a set of practices.
- **Perception:** Process by which people translate sensory impressions into a coherent and unified view of the world around them. Though necessarily based on incomplete and unverified (or unreliable) information, perception is 'the reality' and guides human behavior in general.
- **Predictability:** is the degree to which a correct prediction or forecast of a system's state can be made either qualitatively or quantitatively
- **Procedure**: Fixed, step-by-step sequence of activities or course of action (with definite start and end points) that must be followed in the same order to correctly perform a task. Repetitive procedures are called routines. See also method.
- **Process**: Sequence of interdependent and linked procedures which, at every stage, consume one or more resources (employee time, energy, machines, money) to convert inputs (data, material, parts, etc.) into outputs. These outputs then serve as inputs for the next stage until a known goal or end result is reached.
- **Repeatability:** is the variation in measurements taken by a single person or instrument on the same item and under the same conditions. A measurement may be said to be *repeatable* when this variation is smaller than some agreed limit.
- **Restrictions**: Limitations which cannot be exceeded or rules which cannot be broken. These can be either explicit or implicit.
- **Standard procedure**: a prescribed procedure to be followed routinely
- **Strategy**: Alternative chosen to reach a desired future, such as achievement of a goal or solution to a problem.
- **System**: a set of interdependent dynamic entities forming an integrated whole, which search to achieve an objective by manipulating data, energy, and/or matter.
- **Technique**: Systematic procedure, formula, or routine by which a task is accomplished.
- **Technology**: Purposeful application of information in the design, production, and utilization of goods and services, and in the organization of human activities. Technology is generally divided into five categories:

    - Tangible: blueprints, models, operating manuals, prototypes.
    - Intangible: consultancy, problem solving, and training methods.
    - High: entirely or almost entirely automated and 'intelligent' technology which manipulates ever finer matter and ever powerful forces.
    - Intermediate: semi-automated 'partially intelligent' technology that manipulates refined matter and medium level forces.
    - Low: labor intensive 'dumb' technology that manipulates only coarse or gross matter and weaker forces.
- **Tool**: An item or implement used for a specific purpose. It can be physical (like mechanical tools) or a technical object (e.g. web authoring tool or software program).