

Computer-Aided Problem Solving - Part 1: Objectives, Approaches, Opportunities

Niccolò Becattini¹, Yuri Borgianni², Gaetano Cascini¹, and Federico Rotini²

¹ Politecnico di Milano, Dip. di Meccanica, Via Giuseppe la Masa, 1, 20156 Milan, Italy

² Università di Firenze, Dip. di Mecc. e Tecn. Ind., Via Santa Marta, 3, 50139 Florence, Italy

Niccolo.Becattini@kaemart.it

Gaetano.Cascini@polimi.it

{Yuri.Borgianni, Federico.Rotini}@unifi.it

Abstract. Among the different aims and scopes of Computer-Aided Innovation (CAI) systems a relevant topic is the support of inventive problem solving tasks. The paper presents the research activity developed by the authors in this domain, encompassing the review of the distinctive features of problems encountered by designers and the common approaches employed to overcome them. A further thread of the investigation carried out in this paper concerns the limitations of computer-based approaches exploiting acknowledged models for problem solving. Downstream of the performed analysis the authors highlight the requirements that a novel CAI application should fulfil, supporting the opportunities for building a dialogue-based system.

Keywords: Computer-Aided Innovation, inventive problem solving, conceptual design, dialogue-based system.

1 Introduction

CAD/CAE systems have reached considerable performances in terms of rapidly carrying out routine tasks, allowing to successfully perform detail design activities. On the contrary, their development has not so far concerned the ability to support the early stages of the design process, such as conceptual design, resulting in limited functionalities to enhance product development cycles. Such design phases, in which designers' creativity and inventiveness are primarily exploited, play a crucial role for the success of the products in the marketplace.

Thus, considering companies' double goal of delivering innovative products and reducing the time to market, CAD/CAE systems have consistently focused just on the second objective, by speeding up the processes concerning detail design. Indeed, starting from the CAD-CAM systems which spread since the '80s, computer means supporting the product development cycle have progressively extended their domain of application, also acquiring virtual prototyping capabilities. Product models correspondingly extended the detail level of representation, in order to allow simplified simulation and testing. In the last decade, CAD/CAE systems have been integrated into PLM (Product Lifecycle Management), further extending their domain

of application, but still missing to support the earlier stages of design. Figure 1 represents the domains of application of these systems with respect to the phases of product development and the abstraction level of the adopted product models.

Thus, the aforementioned scope of shortening the lead times has resulted in less accurate initial design phases, with poor support to problem investigation, function analysis, solutions generation and exploration [1], thus shrinking innovation opportunities.

Computer-Aided Innovation (CAI) systems emerged for suitably assisting the conceptual design phase. As depicted in figure 1, the borders of CAI systems domain are still fuzzy, but their capability to support problem solving seems to be a core element on which the scientific community agrees.

Within this context, the authors' research aims at developing an original computer application viable to overcome the deficiencies of CAD/CAE tools with respect to conceptual design and the limitations of CAI systems in exploiting designers' skills, creativity and intangible knowledge.

Due to space limitations, the recalled subjects are treated in two separate and complementary manuscripts: the current one deals with the identification of the requirements to be fulfilled and the tasks to be performed by a CAI problem solving tool. With reference to these findings, the second paper illustrates a dialogue-based algorithm, embedding the TRIZ logic, tailored for people without any background in problem solving methods. By aiding the investigation of a technical problem, such algorithm represents a first module for a complete CAI system to support the conceptual design phase.

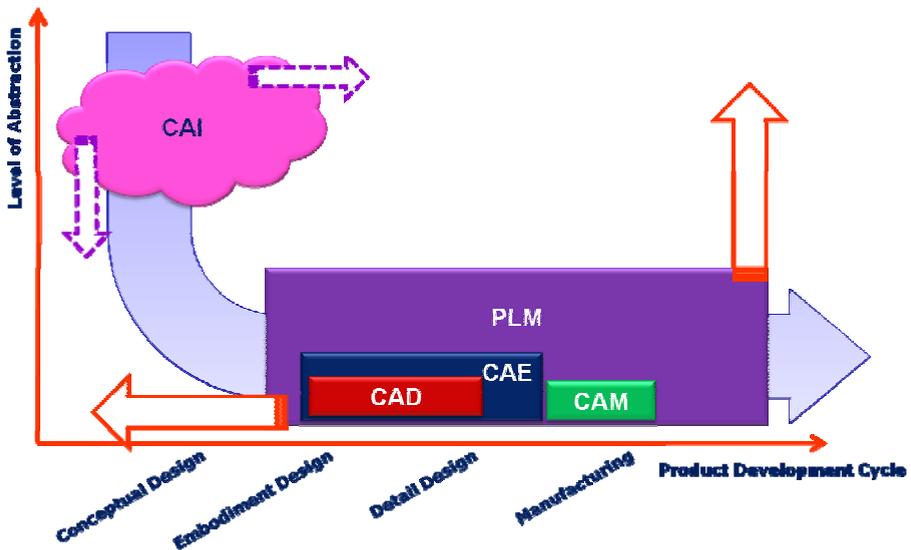


Fig. 1. Current domains of application of PLM and CAI systems within the product cycle and level of abstraction of the corresponding product models. The arrows represent the expected trends of evolution.

According to these premises, this paper firstly performs a review of problems typically encountered by designers (Section 2); then, problem solving approaches and computer-based approaches for problem solving are discussed, focusing on the role played by knowledge in the design context (Section 3). The investigation within innovation tools in design subsequently focuses on the opportunities represented by dialogue-based systems and the employment of natural language (Section 4). The outcomes of the whole research are finally summarized in Section 5.

2 Types of Problems

In the literature there is a plenty of definitions of the term “invention”: among the others, for the scopes of the present paper, it is useful to mention the followings:

- Patent Law recognizes as inventive a technical solution that is useful, novel (no single prior art reference shows the identical development), and unobvious to a person “skilled in the art”;
- Cavallucci et al. associate the concept of invention to the transfer of knowledge between different fields of application; in [2] it is claimed that “Inventive R&D requires the use of knowledge or know how from a substantially different technical domain”.

The first definition is here assumed as the reference to identify an invention, since it is more universally accepted, at least in the industrial world. It relates to the usefulness of an invention, meaning that the technical solution is addressed to satisfy a need, which implies to solve an existing and acknowledged problem. Nevertheless, the second definition is relevant for a class of “inventive problems” and requires a specific solving approach.

Technical problems can be indeed distinguished between inventive and non-inventive, whereas demands and cognitive processes make the differences in this classification. Non-inventive problems don’t require any inventive step, thus relate to situations where the desired outcome can be achieved just by means of an optimal adjustment of system parameters. On the contrary, inventive problems are characterized by at least two conflicting requirements that cannot be satisfied by choosing the optimized values for system parameters. Thus, inventive problems require an inventive solution, which means that it’s necessary to solve a contradiction in order to produce a useful, novel and unobvious answer to an emerging demand.

From another point of view “difficult problems”, according to Funke and Fresch [3], have at least one of four specific characteristics that make them harder to solve than “easy problems”:

- Intransparency, due to the ill-definition of the problem itself, some elements required to achieve the solution are not known;
- Complexity, due to the great number of parameters of the technical system(s) and their mutual connections;
- Dynamics, due to either time-dependent characteristics of relevant features, or to the need of achieving the solution under time pressure;
- Polyteley, which means that the problem is characterized by multiple, non-compatible goals.

From a different perspective, it is possible to distinguish between typical and non-typical problems. Both problems can be approached by means of rules and procedures, but the second ones cannot be solved just by means of such methods; their solution requires the support of knowledge provided by the problem solver and the available informational support. An example of typical problem is the calculation of the roots of an equation of the n -th degree: it's required just to know the appropriate formulas (available for equation from first to 4th degree) or a method (e.g. Ruffini-Horner) that provides the desired approximation of their values. On the contrary, when the problem is non-typical, even if some solution path is provided, the solution cannot be entrusted just to the method itself. This problem solving process requires elements of knowledge that reside outside the boundaries of the method and its language. To get to the desired outcome it's necessary to leverage personal knowledge as well as personal wisdom.

Finally, Simon, in [4], explores the distinction between a well-structured problem (WSP) and an ill-structured problem (ISP): "... a problem may be regarded as well structured to the extent that it has some or all the following characteristics:

1. There is a definite criterion for testing any proposed solution, and a mechanizable process for applying the criterion;
2. There is at least one problem space in which can be represented the initial problem state, the goal state, and all other states that may be reached, or considered, in the course of attempting a solution of the problem;
3. Attainable state changes (legal moves) can be represented in a problem space, as transitions from given states to the states directly attainable from them (...);
4. Any knowledge that the problem solver can acquire about the problem can be represented in one or more problem spaces;
5. If the actual problem involves acting upon the external world, then the definition of state changes and of the effects upon the state of applying any operator reflect with complete accuracy in one or more problem spaces the laws (laws of nature) that govern the external world;
6. All of these conditions hold in the strong sense that the basic processes postulated require only practicable amounts of computation, and the information postulated is effectively available to the processes – i.e., available with the help of only practicable amount of search...."

His description is founded, by his own admission, on a set of features that problems must have in order to be considered as WSP, since it's impossible to provide a formal definition. The class of ISP, indeed, must be considered as a residual of the one related to WSP.

Given these commonly faced situations, Computer-Aided Innovation (CAI) systems constitute an emerging technology for assisting the conceptual design phase by supporting the solution of inventive problems, but also improving the efficiency of any problem solving activity arising in an innovation process. A framework for Computer-Aided systems should be thus capable to face and solve:

- inventive problems by the search of conflicting requirements and the identification of features that the technical solution should have;
- difficult problems by both clarifying their definition and prioritizing the objectives;

- non-typical problems by helping the user in retrieving useful data and information from various domains;
- ill-structured problems by organizing the designer's knowledge according to a framework useful for a successful solution.

These objectives are achieved if the framework helps to approach the problem with a structured process that, during the analysis, stimulates the user's knowledge and his/her creative skills. Thus, a problem solving tool should have the capability of assisting the user in the exploitation of his/her competences during the solving process. Moreover, when the solution space depicted by the designer doesn't provide any successful result, the CAI system should drive the user in the widening of its design space by enriching it with external knowledge. Relevant information should be first searched in a close field of application, since also experts not necessarily fully master their own domain; then, the search should be extended to completely different domains of technical knowledge where similar problems have been already solved with success. With the goal of maximizing the efficiency of the innovation process, such an information search must be performed as automatically as possible by the computer system and should require minimal efforts to the designer.

3 Problem Solving

This section introduces some key concepts related to inventive problem solving and conceptual design, with references to approaches, techniques and research insights relevant for the present work.

3.1 Problem Solving Approaches

The emerging demands for better performances, the reduction of resources consumptions and the removal of harmful side effects generate technical problems due to the conflicting nature of these features. Whenever the optimization of the values of such parameters allows to satisfy the design demands within the established constraints, the solution doesn't require any inventive activity. On the contrary, whenever two or more requirements (demands or constraints) appear as non-mutually compatible just by changing the values of the related design parameters, the issue requires a dialectical approach in order to redesign the system. This inventive problem task implies that the new solution (synthesis) must integrate the needed features (thesis and antithesis) after the removal of the causes that prevent their direct implementation. Optimization and trade off could be considered as routine design, while inventive problem solving relates to non-routine design [5].

Non-routine design activity requires a creative contribution for problem solving. The creativity leaps underneath the solving process have been deeply studied since the '70s [6-8] both to understand human thinking and to provide an efficient way to improve the problem solving activity. With a particular emphasis in [4], Simon distinguishes between ill-structured and well-structured problems and observes that the problem solving approach should be the same, regardless of the problem structure. In a recent paper [7], Dorst calls into question the differences claimed by Simon between well-structured and ill-structured problems, also by highlighting that those

differences mainly reside in the skills of the problem solver of turning the latter into the former. Therefore, even according to Simon, the designer's subjectivity becomes relevant for the design process, since the greatest part of its creative contribute is addressed in changing the structure of the problem. To this end, particular regard should be given to the problem solver's interpretation of the problem, both taking into account his knowledge and the method he follows. Kruger and Cross [8] recognize four different types of cognitive strategies in design:

1. Problem driven design: the designer focuses closely on the problem at hand and only uses information and knowledge that is strictly needed to solve the problem. The emphasis lies on defining the problem.
2. Solution driven design: the designer focuses on generating solutions, and only gathers information that is needed to further develop a solution. The emphasis lies on generating solutions.
3. Information driven design: the designer focuses on gathering information from external sources, and develops a solution on the basis of this information.
4. Knowledge driven design: the designer focuses on using prior, structured, personal knowledge, and develops a solution on the basis of this knowledge. Only minimal necessary information from external sources is gathered.

The solution, whatever the adopted strategy, must be found by means of the creative leap of the problem solver or even by increasing his/her knowledge by information gathering. Beyond the cognitive aspects, the above strategies are characterized by different approaches to systematize the design process, then it is worth to make a brief distinction between cognitive and systematic features of the problem solving methods.

Cognitive approaches are focused on creative thinking features like analogy, abstraction and references to previous experiences by associations of ideas. Furthermore, they can be used regardless of the technical/industrial domain and the increase of their effectiveness must rely on multidisciplinary working teams composed by creative people. These methods leverage tacit knowledge, stimulate "cross-fertilization" thinking processes and individual creative attitude upon appropriate conditioning techniques [9].

On the other hand, systematic approaches of problem solving are characterized by linear and ordered "step-by-step" procedures that drive the design process. They rely on explicit knowledge such as information and data available in handbooks, patents, and scientific literature. Furthermore, both teams of experts and individuals can use them since the creative attitudes are welcome, but not strictly necessary. One of the greatest restrictions of these methods stands in their limited versatility: they are suitable just for specific kinds of problems and are hard to be generalized for different expertise domains.

Despite many creative process models and techniques might be considered, e.g. those reviewed in [10], the discussion is here limited to four well known problem solving methods because they allow to highlight a complete set of relevant issues for the present research. Let's examine in detail the main features, differences and weak points of two methods mostly based on the cognitive aspects of idea generation (Brainstorming and CBR, Case-Based Reasoning) and two methods relying on

systematic procedures for generating a solution (CPS, Constraint Satisfaction Problem and TRIZ, Theory for Inventive Problem Solving).

Brainstorming is a well known method for generating a flood of new ideas: a group of open-minded and creative experts in various domains elicits ideas and thoughts to solve the given problem during sessions, thus triggering off new ideas in the mind of other participants. This method is strongly based on memory stimulation and on the association of ideas from different knowledge domains. Pahl and Beitz [11] suggest the use of such method whenever the design phase comes to a deadlock because of the lack of practical solution concepts and physical principles to be exploited to get the desired outcomes. During a brainstorming session the working team may produce a huge number of ideas even if the greatest part of them is far from being technically or economically feasible. The greatest drawback of this method stands in its implicit trial and error approach: the generated ideas require appropriate validation and the efficiency of the whole innovation process results to be quite poor. Moreover, a brainstorming session intrinsically leverages only the knowledge of the individuals involved in the idea generation.

The aim of Case-Based Reasoning for problem solving is to build a new solution for a given problem by adapting solutions of past cases [12; 13]. This method makes use of cognitive resources with a different approach, since it leverages an archive of successfully solved cases. Despite its cognitive nature, CBR relies on a well structured procedure described by some authors as the “4 R”: “Retrieve, Reuse, Refine and Retain”. During the retrieval phase the knowledge base of successfully solved cases is searched by exploiting an indexing system that summarizes relevant features of the cases themselves. The best solution gets chosen by means of analogy and similarity criteria. The Reuse phase associates the solution of the retrieved case to the new problem, while the refinement phase checks the overall validity of the proposed solution by verifying whether any modifications are required. The retain phase starts whenever the solution has been adapted to the problem: the new case, comprising the initial problem and its solution, becomes part of the knowledge base of the CBR system.

Constraints Satisfaction Problems (CSP) is a method for finding suitable values for variables subjected to constraints on acceptable combinations of values [14]. A CSP model consists of a set of variables, a set of possible values for each variable and a set of constraints the variables must satisfy. Whenever these constraints cannot be completely satisfied, the user has to deal with an over-constrained problem, an optimization problem that requires non-routine design since its solution cannot be achieved by means of values adjustment. Several CSP methods for over-constrained problem are available to solve non-routine design problems such as Constraint hierarchies, Partial CSP and Extending CSP (Probabilistic CSP; Fuzzy CSP; Weighted CSP). All these methods can be classified according to two general strategies:

- those (as Extending CSP, Constraint hierarchies) that allow to state preferences about constraints, choosing the ones to be overridden whenever the problem model is over-constrained. The CSP algorithm tries to solve the problem by satisfying all the constraints; otherwise the solution is searched by satisfying as many constraints as possible, fulfilling all the required or “hard” ones and neglecting the ones marked as preferential or “soft”.

- those (as Partial CSP) that allow to relax the boundary values of variables and constraints whenever an over-constrained problem arises. Nevertheless these methods don't allow the introduction of new variables in the problem model, reducing the space of potential solutions.

As well known by TRIZ practitioners, the theory developed by Altshuller [15] has several models to describe problems, technical systems, solutions and to address the problem solving process, which is systematically driven according to its main classical tool, the Algorithm of Inventive Problem Solving (ARIZ). ARIZ is a step-by-step procedure that integrates all the others TRIZ instruments and brings from the analysis of two contradictory requirements to the synthesis of a new technical system capable of fulfilling such goal. From a formal point of view, this method cannot be considered as completely systematic since “ARIZ is a tool to aid thinking, but it cannot replace thought itself, (...) if the human brain does not use the power of a lifetime’s knowledge, a lot of potential associations and images would be neglected. But it is exactly the sparks of imagination that lead humans to produce creative non typical ideas” [16].

The above-described features of these four methods are depicted in Figure 2, 3 and 4 by means of boxes. Each box qualitatively places the method in a two dimensional diagram, where each axis is referred to a specific characteristic. The boxes may have continuous or dashed border meaning, respectively, that the described characteristic is explicitly presented in the reference or has been inferred by the authors of this paper. Whenever a specific feature of a method cannot be certainly classified because of its intrinsic variability the related box gets wider.

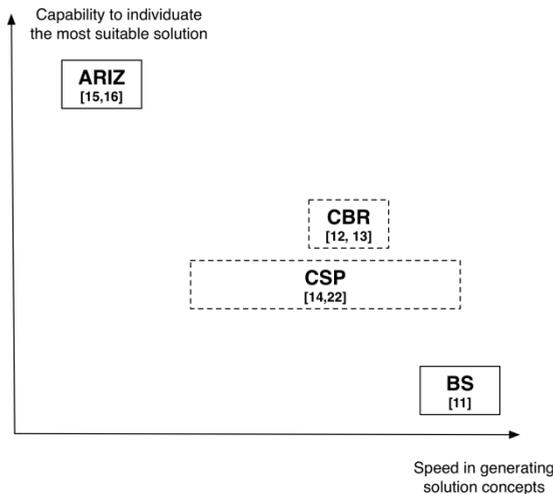


Fig. 2. Brief synthesis of the efficiency of problem solving methods: in this diagram resources of time needed to produce solution concepts are compared to the engineering value of the same concepts

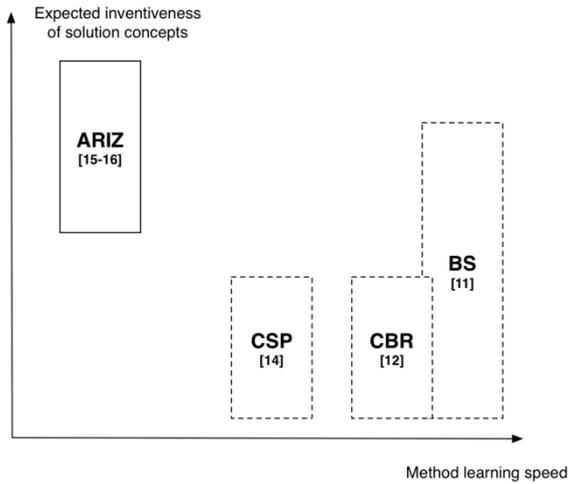


Fig. 3. Qualitative comparison between the capability of solving problems by solution concepts of great inventiveness (whose definition is available in [15]) together with the time required to learn and proficiently use the method itself

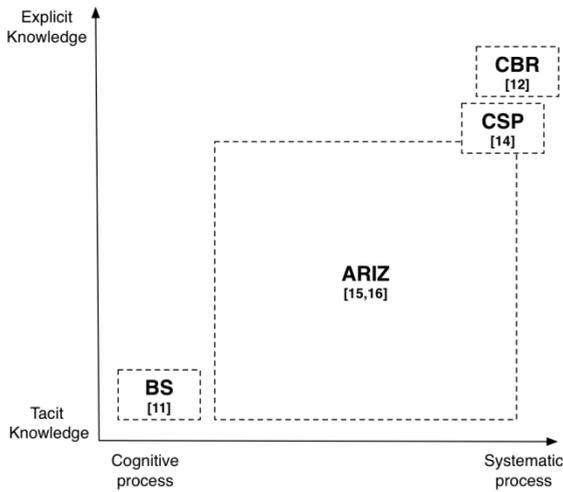


Fig. 4. Problem solving process, differences among the four methods. Cognitive processes leverage tacit knowledge and, on the other hand Systematic processes exploit explicit knowledge. Nevertheless, meta-cognitive methods with prescriptive procedures leverage both type of knowledge.

Both cognitive and systematic methods of problem solving have strong and weak points. The importance of individual knowledge is crucial in problem solving activities, and then it's important to combine the power of systematic approaches, in order to overcome with efficient processes the boundaries of personal creativity, with the capability of cognitive methods to leverage individual tacit knowledge.

3.2 Computer-Aided Systems for Problem Solving and Conceptual Design

The domain of Computer Aided Innovation (CAI) includes systems aimed at assisting Inventive Problem Solving by stimulating creativity and guiding towards suitable problem solving paths.

In the last decade, Information Technology systems have substantially fostered a shared vision of creative patterns among different disciplines, resulting in a consistently growing interest in creativity concept [17]. This led towards the birth of a novel and fertile field of research, namely the interplay between creativity stimulation and computer systems. Given the development of software systems that support the human creativity, Lubart [18] eventually proposes a classification among the ways such aid is provided, with a growing degree of machine involvement:

- by facilitating the management of the working process, encouraging the perseverance of designer in the research of innovative solutions;
- by easing the communication between design team members, since circulation and integration of ideas play a relevant role in the creative process;
- by aiding the designer with a coaching activity, acting as an expert system that guides the user throughout cognitive processes;
- by cooperating in the creative process, thanks to the Artificial Intelligence systems that contribute to ideas generation.

Across all kinds of computer supports for the exploitation of creativity, some emerging issues are argued about the development and the requirements of these systems, thus including those mostly addressed to conceptual design and problem solving. Within computer based environments, Hewett [19] remarks that creative key conditions and processes are not domain specific, as well as the need is stressed for analysing different facets of the problem under different perspectives. Shneiderman [20], performing a review of well-acknowledged design principles to support creativity, highlights the importance of outlining the steps that led to partial solutions in order to maintain comparative criteria for evaluating different options.

It is beyond the objective of this manuscript to provide a complete state of the art of principles and hints that are suggested for successful creative CAI tools. However, the issues that have been briefly introduced provide a common background and a sufficient support in order to reveal inadequate features of the software systems that have implemented the four problem solving methodologies described in the Section 3.1, regardless of the predominance of cognitive or systematic aspects. The CAI applications with a brainstorming framework reflect the recalled lacks of the methodology in selecting the most promising solutions due to the lack of assessment criteria; nevertheless in [21] it is claimed that several cognitive methods, including Brainstorming, have been implemented into software tools. Beyond their preferential development as Artificial Intelligent systems for help desks and customer services, CBR tools are characterized by being strongly domain specific. By their definition and way of working, CPS systems relentlessly lead to the embodiment of a trade off among various requirements, since they propose only partial resolution of the problem by constraints relaxing [22]. Regarding TRIZ based CAI tools, classified among the most diffused software systems for idea generation [21, 23], their main limitation stands in the implementation of a “generic scheme (almost algorithmic) for designing

new product or concept from existing ones”, providing “inventions of only incremental nature” [24]. Thus the richness characterizing the systematic problem solving paths of TRIZ gets lost together with the abstraction capabilities and the range of aspects otherwise depicted. Also other authors [25] judge unsatisfactory the performances of currently available TRIZ software, appointing the attention to a closer link with other design tools and to role played by “dialectical thinking in inventive problem solving and innovation”. Figure 5 shows how much the abovementioned methods have been implemented into software tools together with their adaptability to case studies in different fields of technique.

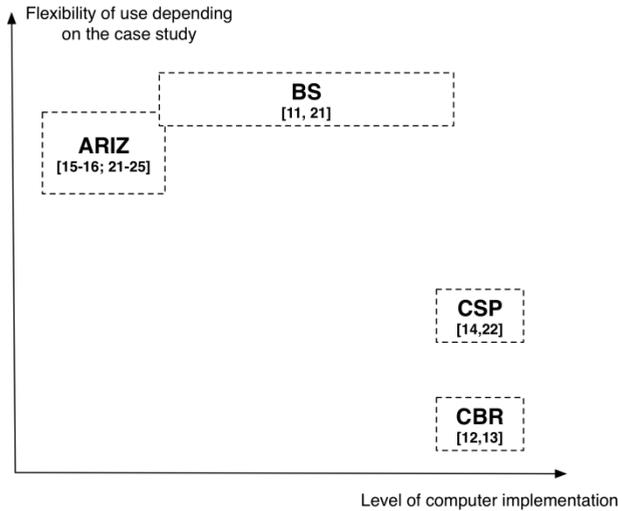


Fig. 5. Flexibility of the four problem solving method briefly presented in the paper, compared to their level of integration in computer-aided tools

3.3 The Role Played by the Information Gathering Task within Conceptual Design

The literature clarifies how information gathering currently requires a meaningful amount of time spent by designers, especially during the conceptual design stage. At the same time engineering designers, especially those with limited experience, are not always aware of the information they require [26]. Interpersonal communication emerges as a key resource for progressing design problems in a practical way: designers generally prefer to source knowledge and information through informal interactions with their colleagues [27].

However, it is expected that designers will increasingly have to rely on retrieving information captured and stored independently of human memory. These reasons provide compelling evidence about the need, for computer-aided problem solving frameworks, to quickly and correctly formulate queries for the investigation of knowledge databases. As a consequence, with the aim of speeding up the search for valuable information, particular attention has to be paid to the analysis of the

encountered problem, so that the main criticalities are individuated, as well as the most characterizing technical parameters, features, elements of the system. A CAI tool for supporting conceptual design therefore requires to guide the user in an accurate and systematic examination of the problem to be faced, clarifying the scopes and the priorities in the solution search, especially in the cases characterized by multiple tasks, complex situations, and tangled interrelations among parameters, effects and physical phenomena. The utmost aim should be to provide the designer, downstream the situation analysis, a limited number of solutions, characterized by promising development and great feasibility.

4 The Opportunities Offered by Dialogue-Based Systems

Within the methods supporting conceptual design with an intensive human involvement, which are currently deemed to be more reliable [28], a dialogue-based system represents a chance for embodying an application to support inventive problem solving in a CAI system. The human-computer dialogue field is claimed to be a rather developed technology and questionnaires administered in a digital form are widespread and increasing. The strong points of computer based dynamic questionnaires lie upon their speed in being carried out [29] and suitability within information retrieval systems [30].

The removal of specific jargon to support users without wide vocational experience, regardless the theoretical support to be chosen for a novel CAI application, encourages to develop a dialogue by the employment of natural language. The literature clarifies that natural language processing is widespread and employed in different engineering domains, as reviewed in [31]. The research opportunities emerging within design field by linking natural language to engineering and technical concepts [32], resulted in an increased interest on questioning techniques. Eris [33] has thus introduced new question categories within well-established taxonomies, by observing the nature of commonly asked concepts within design process. The observations he had carried out gave rise to Generative Design Questions, i.e. explorative queries representing the designer's diverging way of thinking, that investigate possible outcomes and embodiments of a system, as well as the ways to achieve the desired outputs.

As a consequence of the developing research field, the attempts of applying Question-Answer techniques within design process are quite diffused. Wang and Zeng [34] have elaborated an iterative question asking method with the aim of correctly identifying the requirements of new products on the basis of customer surveys. Andersson [35] employs a Question-Answer tool to be employed in the final design stages, such as the verification of the product requirements. Less efforts have been paid towards the employment of such tools within conceptual design.

A particular branch of question-answering sequences in the field of conceptual design has been developed to support Conversational Case-Based Reasoning (CCBR), an extension of the Case-Based Reasoning paradigm [36]. The most relevant advantages of CCBR are related to its capability to incrementally elicit the problem formulation through the interactive dialogue with the user [37], as well as their

flexibility of use and the ease of implementation [38]. Unfortunately their employment domains rarely overlap with engineering design.

The dialogue based system, that is individuated as a chance for a proper CAI system, should thus strive to fulfil the benefits highlighted in the brief overview of Question/Answer techniques, such as effectiveness, simplicity, ease of implementation, capability of following the divergent thinking process inherent to design.

The authors believe that, through a dialogue-based system undertaking the abstraction process, a systematic succession of questions is viable to support the investigation of the problem, allowing even to highlight the least codified aspects, that otherwise would be hardly taken into consideration. The questioning procedure could advantageously foresee differentiated set of queries focusing on various facets of the problem. The logical succession can thus guide the user towards the identification of criticalities and conflicting issues and subsequently to the search of useful technical and scientific information.

5 Conclusions

The paper has investigated and highlighted the main requirements that a novel CAI system should fulfil in order to support inventive problem solving activities within the conceptual design stage. The findings of the present research can be thus summarized in the following major capabilities that should characterize innovative software applications:

- to face inventive, difficult, non-typical and ill-structured problems;
- to overcome the dichotomy between cognitive and systematic problem solving models, in order to both exploit user's knowledge and guide the design activity towards the solution with a step by step process;
- to allow a wide description of the problem, that enlarges the potentialities of solutions from different expertise domains;
- to provide comparative means among the feasible solution concepts;
- to link the design process with valuable external sources of information;
- to take advantage of the opportunities given by Question-Answer techniques in terms of such effectiveness, simplicity, ease of implementation;
- to use natural language in order to assist users with poor scientific background about well-established problem solving methods and with the aim of assisting the designer's thinking process.

As recalled in the Introduction, such features are taken into account in the complementary paper [39], that illustrates an original algorithm proposed by the authors to aid the investigation of an engineering problem.

References

1. Tay, F., Gu, J.: Product modeling for conceptual design support. *Comput. in Ind.* 48, 143–155 (2002)
2. Cavallucci, D., Rousselot, F., Zanni, C.: Assisting R&D activities through definition of problem mapping. *CIRP J. Manuf. Sci. Technol.* 1, 131–136 (2009)

3. Funke, J., Frensch, P.A.: Complex problem solving: The European Perspective. In: Jonassen, D.H. (ed.) *Learning to Solve Complex Scientific Problems*, pp. 25–47. Lawrence Erlbaum, New York (2007)
4. Simon, H.A.: The structure of ill-structured problems. *Artif. Intell.* 4, 181–201 (1973)
5. Gero, J.S., Maher, M.L.: *Modeling Creativity and Knowledge-Based Creative Design*. Lawrence Erlbaum Associates Inc., Hillsdale (1993)
6. Newell, A., Simon, H.A.: *Human problem solving*. Prentice-Hall, Englewood Cliffs (1972)
7. Dorst, C.H.: Design Problems and Design Paradoxes. *Des. Issues* 22, 4–17 (2006)
8. Kruger, C., Cross, N.: Solution driven versus problem driven design: strategies and outcomes. *Des. Stud.* 27, 527–548 (2006)
9. Ross, V.E.: A model of inventive ideation. *Think. Skills Creat.* 1, 120–129 (2006)
10. Howard, T.J., Culley, S.J., Dekoninck, E.: Describing the creative design process by the integration of engineering design and cognitive psychology literature. *Des. Stud.* 29, 160–180 (2008)
11. Pahl, G., Beitz, W.: *Engineering Design - A Systematic Approach*, 2nd edn. Springer, London (1996)
12. Lopez de Mántaras, R., Plaza, C.: Case-based reasoning: an overview. *AI Commun.* 10, 21–29 (1997)
13. Aamodt, A., Plaza, C.: Case-based reasoning: Foundational Issues, Methodological Variations and System Approaches. *AI Communications* 7(1), 39–59
14. Kumar, V.: Algorithms for Constraint-Satisfaction Problems: A Survey. *AI Magazine* 13(1), 32–44 (1992)
15. Altshuller, G.S.: *Creativity as an Exact Science*. Gordon and Breach Science Publisher, New York (1984)
16. Khomenko, N., De Guio, R., Lelait, L., Kaikov, I.: A Framework for OTSM-TRIZ Based Computer Support to be used in Complex Problem Management. *Int. J. Comp. Appl. in Technol.* 30, 88–104 (2007)
17. Candy, L., Hori, K.: The digital muse: HCI in support of creativity, creativity and cognition comes of age. *Interact. J.* 10, 44–54 (2003)
18. Lubart, T.: How can computers be partners in the creative process: Classification and commentary on the Special Issue. *Int. J. Hum.-Comput. Stud.* 63, 365–369 (2005)
19. Hewett, T.T.: Informing the design of computer-based environments to support creativity. *Int. J. Hum.-Comput. Stud.* 63, 383–409 (2005)
20. Shneiderman, B.: Creativity support tools: accelerating discovery and innovation. *Commun. ACM* 50, 20–32 (2007)
21. Hüsig, S., Kohn, S.: Computer aided innovation—State of the art from a new product development perspective. *Computers in Industry* 60, 551–562 (2009)
22. Dubois, S., Rasovska, I., De Guio, R.: Comparison of non solvable problem solving principles issued from CSP and TRIZ. In: *IFIP 20th World Computer Congress, Proceedings of the Second Topical Session on Computer-Aided Innovation. WG 5.4/TC 5 Computer-Aided Innovation*, pp. 83–94. Springer, Boston (2008)
23. Shai, O., Reich, Y., Rubin, D.: Creative conceptual design: Extending the scope by infused design. *Comput.-Aided Des.* 41, 117–135 (2009)
24. Mitra D.: Computational Creativity: Three Generations of Research and Beyond. In: *AAAI Symposium on Creative Intelligent Systems*, Stanford, United States of America (March 26-28, 2008)
25. Leon Rovira, N.: The future of computer-aided innovation. *Comput. in Ind.* 60, 539–550 (2009)

26. Ahmed, S.: An Approach to Assist Designers with their queries and designs. In: Proceedings of ASME 2006 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE 2006), pp. 359–366. ASME, New York (2006)
27. Aurisicchio, M., Bracewell, R., Wallace, K.: Understanding how the information requests of aerospace engineering designers influence information-seeking behaviour. Accepted for publication on *J. Eng. D* (2009)
28. Rohde, D., Pavković, N., Aurisicchio, M.: Design knowledge indexing in relational database. In: Proceedings of the 10th International Design Conference on DESIGN 2008, pp. 857–864. University of Zagreb, Zagreb (2008)
29. Quarteroni, S., Manandhar, S.: Designing an interactive open-domain question answering system. *Nat. Lang. Eng.* 15, 73–95 (2008)
30. Hardré, P.L., Crowson, H.M., Xie, K., Ly, C.: Testing differential effects of computer-based, web-based and paper-based administration of questionnaire research instruments. *Br. J. Educ. Technol.* 38, 5–22 (2006)
31. Chiu, I., Shu, L.H.: Biomimetic design through natural language analysis to facilitate cross-domain information retrieval. *Artif. Intell. for Eng. Des. Anal. Manuf.* 21, 45–59 (2007)
32. Zeng, Y.: Recursive object model (ROM) - Modelling of linguistic information in engineering design. *Comput. in Ind.* 59, 612–625 (2008)
33. Eris, Ö.: Asking Generative Design Questions: a Fundamental Cognitive Mechanism in Design Thinking. In: Folkesson, A., Gralen, K., Norell, M., Sellgren, U. (eds.) Proceedings of the 14th International Conference on Engineering Design (ICED 2003), Stockholm, pp. 587–588 (2003)
34. Wang, M., Zeng, Y.: Asking the right questions to elicit product requirements. *Int. J. Comput. Integr. Manuf.* 22, 283–293 (2009)
35. Andersson, K.: An iterative Question-Answer driven process to support product design. In: Lehtonen, T., Pulkkinen, A., Riitahuhta, A. (eds.) Proceedings of NordDesign 2004, Tampere, pp. 173–182 (2004)
36. Aha, D.W., Muñoz-Avila, H.: Introduction: Interactive Case-Based Reasoning. *Appl. Intell.* 14, 7–8 (2001)
37. López de Mántaras, R., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M., Cox, M., Forbus, K., Keane, M., Aamodt, A., Watson, I.: Retrieval, reuse, revision and retention in case-based reasoning. *Knowl. Eng. Rev.* 20, 215–240 (2006)
38. Aha, D.W., Breslow, L.A., Muñoz-Avila, H.: Conversational Case-Based Reasoning. *Appl. Intell.* 14, 9–32 (2001)
39. Becattini, N., Borgianni, Y., Cascini, G., Rotini, F.: Computer-Aided Problem Solving - Part 2: a Dialogue-Based System to Support the Analysis of Inventive Problems. In: Cavallucci, D., De Guio, R., Cascini, G. (eds.) CAI 2011. IFIP AICT, vol. 355, pp. 132–148. Springer, Heidelberg (2011)