INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN

ICED 91

ZURICH, AUGUST 27-29 1991

TOWARDS A DECISION-SUPPORT FRAMEWORK FOR MECHANICAL CONCEPTUAL DESIGN

Amaresh Chakrabarti and Thomas P. Bligh

ABSTRACT

Mechanical conceptual design involves a recursive problem solving activity cycling through problem definition, synthesis of (partial) solutions, solution evaluation, and *(vertical & horizontal)* problem redefinition. A computational framework for supporting conceptual design should have tools, for carrying out the above tasks, with a representation having *domain, level* and *perspective* independence. In this paper, we argue that there is potential for the development of a decision-support framework which is *domain* and *level* independent, if *horizontal* and *vertical* problem redefinition could be integrated, into an object oriented representation, with *perspective-based* evaluation tools.

INHALTSANGABE

Konzeptueller Entwurf in der Mechanik besteht aus eine rekursive problemloesende Aktivitaet, die durch Problembeschreibung, Synthese (partielle) Loesungen, Beurteilung der Loesungen und (senkrechte und waagerechte) Problemneubeschreibung kreist. Ein rechnerunterstuetzte Framework, das den konzeptuellen Entwurf unterstuetzt, sollte fuer die obigen Aufgaben Werkzeuge haben, deren Darstellungen unabhaengige Domaenen, Ebenen und Perspektiven haben. In diesem Artikel argumentieren wir, dass die Entwicklung eines entscheidungsunterstuetzenden Frameworks mit unabhaengigen Domaenen und Ebenen moeglich sein koennte, wenn die waagerechte und senkrechte Problemneubeschreibungen in eine objektorientierte Darstellung zusammen mit Perspektiv-basierte Beurteilungswerkzeuge integriert werden koennen.

INTRODUCTION

The conceptual design phase involves the activity of generating solution concepts to satisfy specified requirements [1]. There is usually more than one concept that will satisfy a given set of requirements. This indicates that there is scope for producing better or even optimal solutions, if the solution space can be explored adequately. However, the bias of designers towards using familiar solutions may prevent them from exploring novel alternatives. Moreover, as the design progresses, the information handled grows larger , rendering data management a major problem. These, along with the constraint of deadlines, limits the search of the solution space, thereby reducing the scope for optimization. An appropriate computational environment holds the promise for reducing these problems and widening the prospect of producing better design solutions.

In this paper, we review existing research to identify the limitations of present computational aids for mechanical conceptual design and their causes; we investigate the nature of problem solving in the conceptual design phase to identify some essential requirements of a useful design environment, and suggest a scheme, for initial research, as a first step towards these long term goals.

THE NATURE OF PROBLEM SOLVING IN CONCEPTUAL DESIGN

Conceptual design involves the generation of solution concepts to satisfy the requirements of the specified requirements. Therefore, the problem description, solution description, and the mappings between the two are crucial to the conceptual design activity. The mapping can be from problem to solution (synthesis), or from solution to problem (evaluation).

In conceptual design, usually the whole problem is not solved in a single step, because:

(a) The problem may not be completely defined. In this case, even if a complete solution is found for the initially ill-defined problem, it would need to be evaluated against the subsequent revised problem definitions as they evolve towards the completely defined problem;
(b) A solution may solve only a part of the problem. In this case, a solution chosen as a candidate being a partial solution, we would need to redefine the original problem while incorporating this partial solution;
(c) The mappings, ie the synthesis and evaluation processes, may not be clearly understood. In this case, we cannot afford to map the complete problem into the solution space in a single step, and hence must try to solve only parts of the whole problem. We then evaluate the solution with respect to the whole problem, and identify what remains to be solved.

The above reasons lead us to view the conceptual design as a recursive activity, as shown if the Fig.1, involving the following steps:

Step 1: A design problem is formulated as an initial problem definition.
Step 2: A part of the problem is chosen for synthesis.
Step 3: A set of solutions is synthesized to satisfy the requirements of the chosen part of the problem.
Step 4: The first chosen solution is evaluated with respect to the complete initial problem, which is then revised, incorporating this solution.
Step 5: Solutions are now synthesized for the unsolved part of the revised problem, the first of which is evaluated to produce the next revised problem definition. This recursive process (steps 2 to 5) is continued until the problem is completely solved.
Step 6: We go back to the intermediate preceding node (little ellipses in Fig. 2) and recurse for all the remaining solutions.
Step 7: We now go back to the node preceding that in step 6 and continue, until the solution tree is searched completely.

The two essential features characterising problem solving in this view of conceptual design, are therefore identified as:
(a) Divide and rule approach:- The problem is not solved as a whole, but tackled in parts.
(b) Problem Redefinition:- The problem redefinition occurs in terms of the previous problem state, the contribution of the newly chosen partial solution towards solving the problem, and the additional requirements the partial solution imposes.

REQUIREMENTS FOR A USEFUL COMPUTATIONAL FRAMEWORK

From the nature of the tasks involved in the conceptual problem solving activity, we can distil some of the essential requirements for a useful conceptual design environment:

Problem Representation: The environment must provide constructs for adequately describing design problems and ways of abstracting and working on various perspectives and subsets of the problems. This requirement stems from the divide-and-rule approach common to this conceptual design problem solving.

Representation of the Solution Concepts: The environment must provide constructs for adequately describing solution concepts and ways of abstracting and working on various perspectives and subsets of the solutions. This requirement arises as a consequence of the previous requirement, as the objective of working on a specific part of the problem is to associate that to the relevant aspects of the solutions.

Provision for Synthesis: The synthesis tool, given a specified part or perspective of a design problem, should be able to identify the solution concepts in the knowledge base that would satisfy the requirements of the chosen part of the problem. The effectiveness of the tool would depend on the richness of the representation.

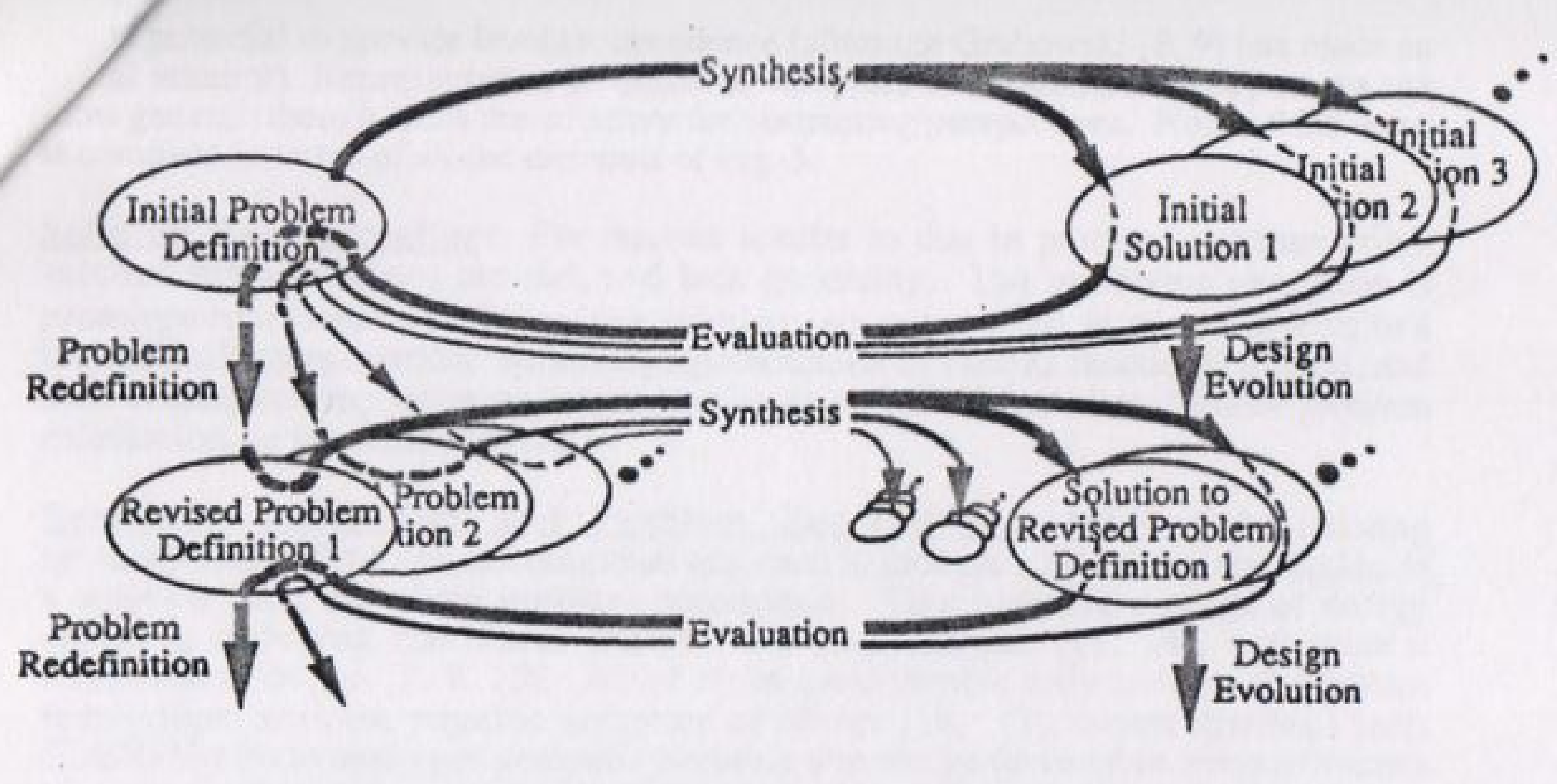

Fig.1 The Nature of Problem Solving in Mechanical Conceptual Design

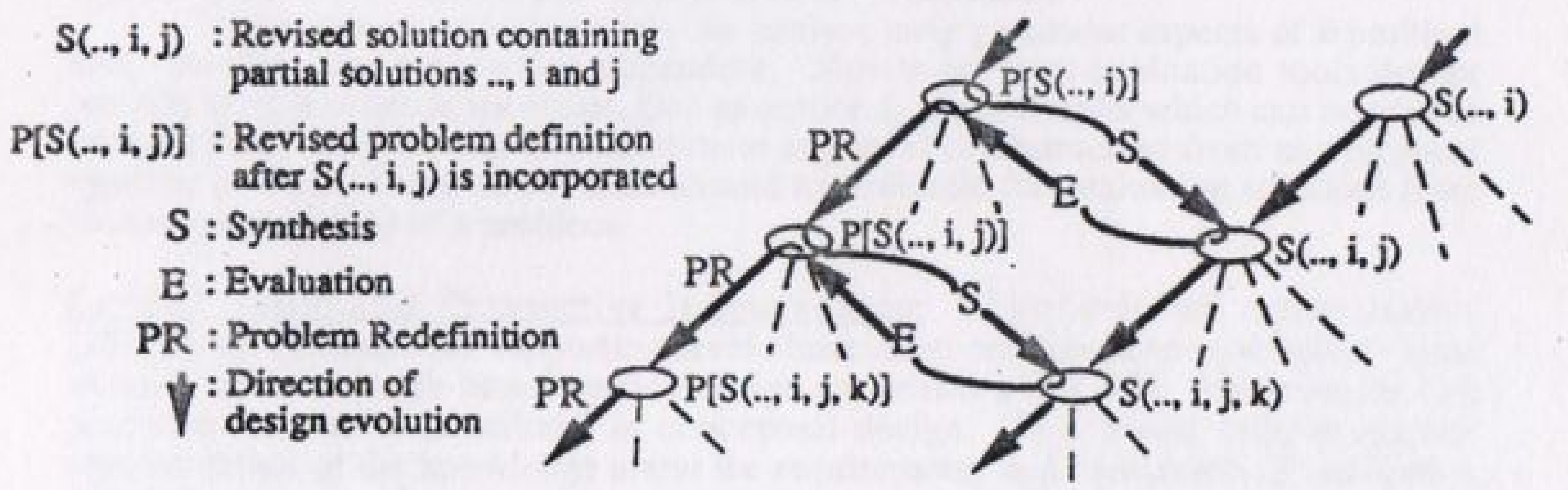

Fig.2 Trees for Problem and Solution States

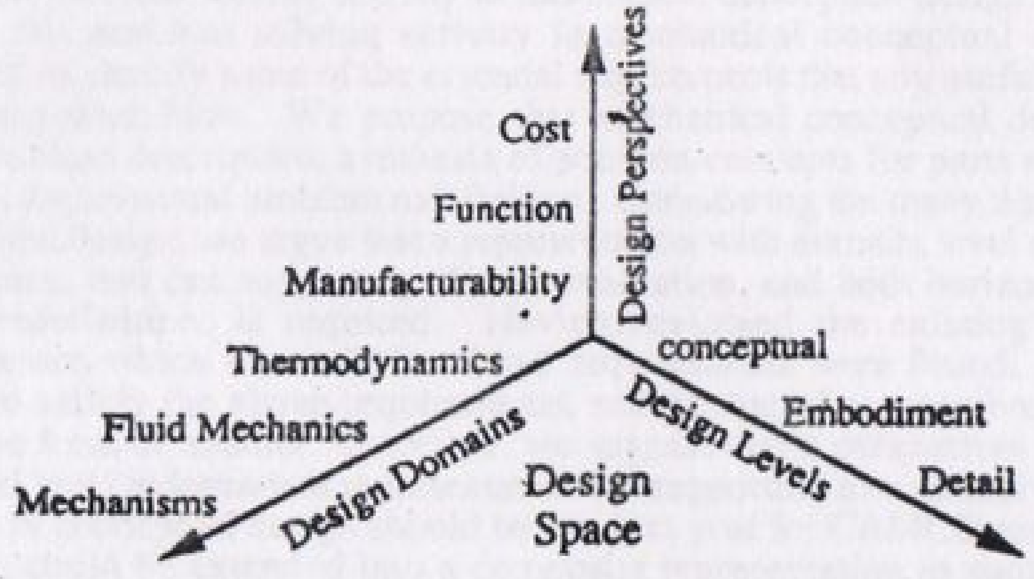

Fig.3 Various Kinds of Abstraction Needed for Design Knowledge Representation

Provision for Evaluation: The evaluation tool, given a specified part or perspective of a design problem and a candidate solution concept, should be able to identify the requirements of the problem that the solution is able to satisfy. This is a must, if problem redefinition is to take place.

Provision for Problem Redefinition: Each partial solution chosen affects the state of the present problem definition, by its contribution to solving part of this problem, and by the requirements it imposes. This requires one to redefine the remaining problem. We term the redefinition, required after only the contribution of a chosen solution to solving a problem is taken into account, a *horizontal problem redefinition*. Suppose, we choose a lever to convert the input torque into a force, as a partial solution to the problem of converting an input torque into an output torque. After horizontal redefinition, the revised problem should be: convert output force from the lever into the specified output torque. On the other hand, the requirements imposed by the chosen partial solution give rise to another set of problems, which can be solved later. We term this a *vertical problem redefinition*. For example, once the horizontal problem solving is complete, we still need to solve for the functional requirements of the lever, such as design of its support, design of a means for the transfer of the lever output, and design of the parts of the lever.

Domain Independence: The representation must be domain independent so that design knowledge from various domains can be integrated within the same framework, see Fig.3. This stems from the fact that mechanical design may involve knowledge from many design domains.

Level Independence: The design knowledge should be representable at various design levels. This is important, as during the design process designers should be able to move freely between the design levels, and the knowledge compiled at various levels should be freely accessed from any level.

Perspective Independence: Perspective independence should ensure that different perspectives of a design problem can be represented by using the same representation. This should also ensure that from the complete problem description, any particular perspective of the problem could be easily abstracted. For example, weight or energy requirements of a transmission system are regarded as specific perspectives of a transmission design problem (see Fig. 3).

LITERATURE SURVEY

The general trend of research in Computer Aided Mechanical Conceptual Design (CAMCD) is to focus on one perspective of one class of problems and to characterize a known class of design solutions in terms of the above perspective. For example, in transmission design (a class of design problems) we could focus on the energy transformations (a perspective) and consider gears (a class of solutions). This method helps to generate a common language between the considered solutions and the particular perspective of the problem, and hence paves the way for devising synthesis and evaluation methods between that problem perspective and the solutions in terms of the common language. We now examine some present literature in CAMCD, in the context of the requirements enlisted in the previous section, and identify their potentials and limitations.

Problem Representation: When developing CAD tools, it is necessary to represent the problems the tool is intended to tackle. Hence, problem representations abound, starting from kinematic representations [2, 3, 4,] to Bond Graphs [5, 6]. However, as the main emphasis has usually been on solving a chosen class of problems from a chosen perspective, the representations are generally restricted to the respective domains of application. Functional representation [7, 1] seems promising in terms of domain independence from a fixed perspective, but has not been adequately researched for its potential to provide level independence (although Grabowski [8, 9] has made an initial attempt). Representation in terms of variables and parameters [10] seems the most general, though lacks the structure for abstracting perspectives. None of the work is complete in terms of all the elements of Fig. 3.

Solution Representation: For reasons similar to that in problem representation, solution representations abound, and lack generality. The promising exception is *prototype representation* [11], where solutions are represented as related objects in a hierarchical representation. However, representation of various functional aspects, and their consequent requirements, are not explicitly stated. This makes flexible problem redefinition, as yet, unattainable.

Synthesis, Evaluation and Problem Redefinition: Most of the existing synthesis tools utilize a transformational approach to produce a functional description of a solution from the given problem description. This includes a range of design domains, including transducer design [12], gear designs [13, 14], and general transmission design [8, 9, 15]. All of these tools involve only horizontal problem redefinition, with the possible exception of Maher [16, 17], whose synthesis tool, EDESYN tries to apply pre-compiled planning knowledge (defined in terms of frames of goals and their sub-goals) to identify the solutions, which are attached as the leaves of these goal frames. This is only vertical problem redefinition, where the goals and their sub-goals are intermediate requirements of the solutions residing at the leaves of the goal frames. What is needed is an integrated approach which would have the capability of both horizontal and vertical problem redefinition.

The existing evaluation tools can analyse only particular aspects of a problem and, therefore, are perspective dependent. Moreover, most evaluation tools do not provide level independence either. One exception is Bond Graphs which can be used to analyse the energy aspects of a solution at any level of abstraction from any physical systems domain. Tools of this kind should be available for evaluating solutions from various perspectives of a problem.

Level, Domain and Perspective Independence: Object oriented representation [18] has the potential for supporting level abstraction, and has been successfully used in building knowledge-based routine design applications [19, 20]. However, its full potential has not been utilized in conceptual design. This would require explicit representation of the knowledge about the requirements and constraints of the known solution concepts. Functional Representation, especially Bond Graphs, is promising for domain independence. However, perspective independence is not available.

CONCLUSION

In this paper, the case that an appropriate computational environment would improve the problem solving activity in mechanical conceptual design is argued. The nature of the problem solving activity in mechanical conceptual design is then investigated to identify some of the essential requirements that any useful computational environment must have. We propose that mechanical conceptual design proceeds through problem description, synthesis of solution concepts for parts of the problem, evaluation, and eventual problem redefinition. Considering the many domains involved in mechanical design, we argue that a representation with domain, level and perspective independence, that can support synthesis, evaluation, and both horizontal & vertical problem redefinition, is required. Having reviewed the existing literature, no representations which meet all the above requirements were found. However, the potential to satisfy the above requirements, except that of perspective independence exist in one form or another. Therefore, we suggest that a perspective dependent and domain and level independent representation for supporting evolutionary synthesis and evaluation in conceptual design should be the next goal for CAMCD research. This, if successful, could be extended into a composite representation to support the design activity for a set of problem perspectives.

REFERENCES

[1] Pahl, G., and Beitz, W. *Engineering Design*, The Design Council, London, 1984.
[2] Freudenstein, F., and Maki, E.R. *Development of an Optimum Variable-Stroke Internal Combustion Engine Mechanism from the Viewpoint of Kinematic Structure*, The ASME Journal of Mechanisms, Transmissions and Automation in Design, Vol.105, June 1983, 259-266.
[3] Hoeltzel, D.A., Chieng, W-H., and Zissimides, J. *Knowledge Representation and Planning Control in an Expert System for the Creative Design of Mechanisms*, AI EDAM, Vol.1, No.2, 1987, 119-137.
[4] Hoeltzel, D.A., and Chieng, W-H. *Knowledge-Based Approaches for the Creative Synthesis of Mechanisms*, Computer Aided Design, Vol.22, No.1, Jan/Feb 1990, 57-67.
[5] Paynter, H.M. *Analysis and Design Of Engineering Systems*, The MIT Press, MA, 1961.
[6] Rosenberg, R.C., and Karnopp, D.C. *Introduction to Physical System Dynamics*, McGraw-Hill, 1983.
[7] Freeman, P., and Newell, A. *A Model for Functional Reasoning in Design*, Proc. Int. Joint Conf. AI, London, England, 1971, 621-633.
[8] Grabowski, H., and Benz, T. *Functional Modelling in Intelligent CAD Systems*, Preprints of the Second IFIP WG 5.2 Workshop on Intelligent CAD, Cambridge, England, 1988, 179-196.
[9] Grabowski, H., and Benz, T. *Implementing the Design Methodology*, Preprints of the Third IFIP WG 5.2 Workshop on Intelligent CAD, Osaka, Japan, 26-29 Sept. 1989, 17-37.
[10] Popplestone, R.J. *The Edinburgh Designer System as a Framework for Robotics: the Design of Behavior*, AI EDAM, Vol.1, No.1, 1987, 25-36.
[11] Gero, J.S., Maher, M.L., and Zhang, W. *Chunking Structural Design Knowledge as Prototypes*, Proc. Third Int. Conf. on the Application of AI in Engg., Amsterdam, 1988, 3-21.
[12] Ulrich, K.T., and Seering, W.P. *Synthesis of Schematic Descriptions in Mechanical Design*, Research in Engg. Design, Vol.1, No.1, 1989, 3-18.
[13] Hoover, S.P., and Rinderle, J.R. *A Synthesis Strategy for Mechanical Devices*, Research in Engg. Design, Vol.1, No.2, 1989, 87-103.
[14] Finger, S., and Rinderle, J.R. *A Transformational Approach to Mechanical Design using a Bond Graph Grammar*, EDRC Technical Report No. 24-23-90, Carnegie-Mellon University, PA, 1990.
[15] Chakrabarti, A., and Bligh, T.P. *Functional Synthesis in Mechanical Conceptual Design*, Submitted for Publication to AI EDAM, 1990.
[16] Maher, M.L. *Engineering Design Synthesis: A Domain Independent Representation*, AI EDAM, Vol.1, No.3, 1987, 207-213.
[17] Maher, M.L. *Synthesis and Evaluation of Preliminary Designs*, Proc. Fourth Int. Conf. on the Application of AI in Engg., Cambridge, UK, July 1989, 3-14.
[18] Su, S.Y.U. *Modelling Integral manufacturing Data with SAM*, IEEE Computer, Jan. 1986, 34-39.
[19] Brown, D.C., and Chandrasekaran, B. *Knowledge and Control of a Mechanical Design Expert System*, IEEE Computer, July 1986, 92-100.
[20] Mittal, S.M., Dym, C.L., and Morjaria, M. *PRIDE: An Expert System for the Design of Paper Handling Systems*, IEEE Computer, Nov. 1985, 102-114.

Authors' Address:
Amaresh Chakrabarti and Dr. Thomas P. Bligh
The Engineering Design Centre, Cambridge University Engineering Department
Trumpington Street, Cambridge CB2 1PZ, U.K.