

N89-25158

New Directions for Artificial Intelligence (AI) Methods
in Optimum Design

Prabhat Hajela
Aerospace Engineering, Mechanics, & Engineering Science
University of Florida, Gainesville, Florida

PRECEDING PAGE BLANK NOT FILMED

Abstract

Developments and applications of AI methods in the design of structural systems is reviewed. Principal shortcomings in the current approach are emphasized, and the need for some degree of formalism in the development environment for such design tools is underscored. Emphasis is placed on efforts to integrate algorithmic computations in expert systems.

Knowledge-Based Systems for Structural Design

Generating an optimum structure is a multistage process, generally initiated with the definition of a structural model to transmit applied loads to the support points. This definition is not necessarily unique and falls in the category of topology assignment problems that are discussed in later sections. Once the topology is known, a finite element analysis is typically invoked, requiring a suitable discretization of the model. This discretized model is used for repetitive analysis in optimum design, a process that is computationally demanding, and very often necessitates the generation of a discrete design model with fewer degrees of freedom. Creation of a design model that is distinct from an analysis model is also often necessary for improving efficiency of optimization algorithms. In addition to creating efficient analysis and design models, the choice of suitable optimization algorithms and algorithm parameters for the problem under consideration, selection of design variables and constraints for the problem, and monitoring as well as enhancing the efficiency of the algorithm, are important steps, requiring insight that is available to few experts in the field.

The need to disseminate such information to a larger user community prompted the development of a prototype expert system OPSYN, configured as an online consultant to provide interactive assistance in the task of finite element modeling, optimum design modeling, and in the selection of optimization strategies for structural design. In addition to a knowledge base and an inference engine with an explanation facility, the system is also equipped with a knowledge-acquisition facility, an input-output facility that includes a knowledge-base editor and a graphical display capability.

The rules for finite element modeling include information pertaining to location of nodes, node numbering, mesh generation,

mesh refinement, element selection, and guidelines to eliminate element distortion. The rules for optimum design modeling include concepts pertinent to selection of design variables, constraints, approximation techniques, and strategies for sensitivity analysis. A third set of rules to enhance optimization performance primarily assists the user in selecting optimization strategies for the design problem. This includes rules for both unconstrained and constrained optimization problems, and rules for algorithm switching in the event that the initial selection is unsatisfactory. Additional details about this implementation are available in References [1-3]. A schematic illustrating the framework of this implementation is shown in Figure 1.

Knowledge acquisition and representation are important issues in expert system development and are addressed in a novel manner in the OPSYN system. Misrepresentation of information is a frequent dilemma, particularly when text alone is used for communication. This limitation was overcome by adding a graphical display of both rules and actions in addition to the textual format. A CAD-based knowledge acquisition system embracing the protocol analysis approach [4], was also developed for this system. Automated knowledge acquisition and analysis to generate new rules are key features of this system.

OPSYN is in the general category of consultative type systems for structural analysis and design modeling (other notable systems include SACON [5], ESSDAN [6] and ADEPT [7]). Such systems facilitate in the dissemination of large amounts of documented and experiential knowledge in this domain to a larger user community. The goals or objectives of the work are well understood, and hence a backward reasoning strategy is naturally applicable. The suitability of such an approach in design, which is essentially a generative process, is questionable. This issue is a principal focus of the present work. The design problem under consideration may be stated as follows: given a design domain, a set of distributed or applied loads, and a set of support points, generate a structural configuration that is optimal with respect to the stated objectives and design constraints.

The traditional design approach starts with a chosen initial configuration, and successively refines this design to obtain an optimum with reference to a set of prescribed constraints. Despite the success in using this approach, it has a limitation in that the final outcome is strongly linked to the initial model abstraction process. Human designers are quite adroit at finding a promising configuration and developing it into a feasible design. The present paper deals with the development and evaluation of strategies to facilitate the automated search for alternative preliminary design configurations, with the goal of spanning the space of conceptually distinct plausible designs.

The preliminary structural layout problem has received some

attention but with no emphasis on optimality of the final design. References [8] and [9] report initial efforts in the development of expert systems for preliminary form design of structural machine parts. The general approach adopted in this study was one in which the structure was decomposed into primitive components. Previously catalogued case studies were used to build the structural form, starting at the primitive level and incrementally adding degrees of complexity. The study also emphasized the need for algorithmic processing in such activity. Nevill and his co-workers [10,11] and Brown [12] also recognize the importance of the preliminary synthesis problem. The research activity in this group has resulted in the development of an automated preliminary design system MOSAIC, currently implemented for 2-D mechanical and structural systems. In this system, point loads are stabilized by connecting them to predefined supports, using structural elements that satisfy other established goals. The thrust of this work is in generating a sequence of promising initial designs, but with no focus on optimality.

The developments described above are drawn from the discipline of structural design. Numerous efforts along similar lines are reported in the chemical, electrical and mechanical engineering domains. These efforts have been somewhat disjointed in form, each devoting significant development effort in building a framework that was considered by its developer as novel and necessary for the application at hand. Other than the fact that such systems have common components in the form of a knowledge base and an inference facility, there is very little adherence to a set of common development guidelines. This unproductive approach results, in most instances, in marginal contributions towards advancing design practice in that domain' and is largely responsible for the growing skepticism about the expert system field in general.

There exists no general agreement on what is regarded as a good model for engineering design practice. In fact, definitions of what design entails are varied, with every practitioner having a different philosophical viewpoint. Recent efforts have focussed at attempting to establish some formalism in knowledge-based design [13,14]. Not surprisingly, the principal concepts of such proposals have been part of the optimum design literature that is traditionally not associated with heuristic methods. Subsequent sections of this paper attempt to develop a framework that is based on one such model of design, largely in the context of optimum structural synthesis.

A General Framework for Automated Structural Design

Tong [13] presents an elaborate description of the principal components of knowledge-based problem solving systems. The framework for automated structural design systems discussed in this section incorporates some of these ideas, and extends them to the structural design domain. At the very outset, it is

important to recognize the levels at which the automated design process can be organized. Tong classifies these as the knowledge level, the function level, and the program level. The present paper describes the attributes of each of these levels of organization, using the task of optimum structural synthesis (both conceptual and refined) as the design domain. A typical relational arrangement of these levels of organization is shown in Figure 2.

Knowledge Level

This level constitutes a rigorous and detailed description of the design domain. Applicable theories for analysis in the domain should be accessible at this level, as should criterion or rules that define acceptable solutions. There is scope for a significant amount of organization at this level. The types of design applications envisaged for such systems would very seldom generate designs that bear no resemblance to their predecessors. It is therefore possible to classify previous feasible solutions based on salient characteristics and the design constraints to which they conform. This step provides a general nomenclature and classification of types of design constraints, of problems that can be solved, and of possible solution strategies.

As stated in earlier discussions, optimum design of structural systems starts with the process of proposing a structural configuration to transmit a set of prescribed loads to given supports. Once an initial model is obtained, it can be refined to yield an optimum. To limit the scope of the present paper, this discussion will be confined to the problem of optimal generation of structural topology. The domains that must be considered in this exercise include geometric modeling, structural analysis, and optimization methodology. The problem is further simplified by restricting load deflection analysis to the linear region. The tools to analyze the structure or parts of the structure must be accessible at the knowledge level. In the implementation under study, a finite element analysis program EAL [15] and several independent analysis modules provide this capability. Also available at this stage are simplistic tools to implement and analyze geometrical layouts of structures for feasibility. Finally, access to gradient and non-gradient based optimization methods is made available at this level [16,17].

In creating a taxonomy of design requirements for the problem under consideration, it is also important to identify any salient characteristics that result from an imposition of such requirements. Structural design requirements may be classified on the basis of strength, stiffness, elastic stability, degree of redundancy, types of support conditions, dynamic behavior, and a requirement of least weight or least complexity in manufacturability. Clearly, each of these requirements has an influence on the design that distinguishes it from designs dominated by other requirements. As an example, a structure that is governed by structural stability requirements will have

elements that can withstand compression (not cables or chains) and further, such elements will typically have aspect ratio and stiffness properties that would reduce elastic instabilities. A structure governed by bending stiffness requirements would have large bending moments of inertia in preferred directions. It is possible that two or more requirements result in similar characteristics, and these must be accounted for in the taxonomy. In as far as possible, however, it is advantageous to distinguish one requirement with one observed characteristic. Failing this, the classification must clearly indicate the relative contribution of a requirement to a salient characteristic.

The class of problems that can be solved is determined by the scope of information in the knowledge level. To further augment the usefulness of a taxonomy based on design requirements, a definition of possible solutions (obtained in previous work) that may be in a primitive or refined form and satisfy the requirements, is proposed in the present work. The stabilization of point loads to supports may be handled by a truss structure. Axial force elements or their combinations resulting in simple truss units are provided at the knowledge level to use for the desired task. These primitive forms may have to be varied to meet current design specifications, an example of which is provided by a manufacturability requirement limiting the length of any one member by lower and upper bounds. These refinements would be introduced at the function level of the design process. In a similar manner, design requirements that require the transmittal of distributed loads to supports, or point and/or distributed loads to a single support, must do so by a beam element or a combination of beam elements. Primitive seed designs to implement this are available at the knowledge level.

There are two additional points about domain knowledge at this level that are very essential to the design process. First, knowledge must be available to judge a proposed solution as an acceptable design. This essentially involves both structural and topological analysis to assess feasibility. The other requirement, and one that is not so easy to satisfy, is the evaluation of the domain theory to see if it contains sufficient knowledge to both generate and recognize a solution to the problem. This has been termed as "epistemological adequacy" by McCarthy [18].

Function Level

The actual task of design implementation is relegated to the function level, as it is desirable to keep all strategies pertinent to the design problem separate from the knowledge level. The design specifications handed down from the knowledge level are attempted to be satisfied at the function level. All problem independent strategies which assist the design process, are confined to this stage. These generic problem solving operators are explained here in context of the structural design problem. A controller must be formulated at this level to direct

the flow of the design from one process operator to another. Although designs can be generated by considering all requirements simultaneously, this methodology is not considered appropriate for the task at hand. Design is, more naturally, a process of refinement in steps to satisfy local goals and to keep track of how the current design step is likely to influence the global design. Some of these approaches are similar to methods of multilevel decomposition represented in recent studies [19,20].

The process of refinement in steps is initiated by a decomposition of the problem into smaller, more tractable, and preferably, single goal problems. The underlying principle in such a refinement is that the solution space is more likely to be unique in the presence of a higher degree of specification detail. The approach is one where a set of refinement operators are invoked by the controller to add greater detail in either the specifications or to the initial design (Figure 3). These two approaches for refinement have their accompanying ramifications, and are discussed in the following paragraphs. The design of a portal framework of beam elements to carry point forces and moments (Figure 4) is used to illustrate these concepts. The choice of the beam cross section must be made between an I-section, an open C-section, and a hollow circular section. The support points are defined and a choice of pin or clamped supports is available.

Refinement in specifications requires that the design specifications be arranged by a priority derived from their relative importance. For the sake of illustration the design specifications for the portal framework are ordered as follows:

- a) The structure must be such that all loads have a load path to all supports.
- b) The structure must not allow static displacements larger than specified values at points of load application.
- c) The components (beam elements) must not be overstressed beyond elastic limits.
- d) Local buckling or crimping in structural elements is disallowed.

The first requirement is of a topological nature and is handled by accessing domain knowledge available at the knowledge level. A controller would invoke an element generating program to generate beam elements that would meet this specification without attention to any other requirements. The next set of specifications would require assignment of cross sectional properties (cross sectional areas and moments of inertia) to the beam elements. No attention is paid at this level to the specificity of cross sections involved. The controller can either look for existing designs at the knowledge level or proceed with a generate-and-test strategy to implement the requirement. The next two requirements similarly dictate selection of particular types of cross sections based on the load conditions and also require detailed sizing of these cross

sections for the problem at hand.

A design obtained by this approach is likely to vary with different ordering of design specifications. This difficulty can be alleviated to some extent by requiring that each design specification be only partly satisfied as it is considered. This is akin to maintaining a constant buffer in the constraint activity and tightening all constraints after each specification has made a contribution to the design.

A second approach of refinement is one where the system design specification is decomposed into design specifications for components of the system. This type of refinement can actually be embedded into the one described above, wherein each design specification is further decomposed into component specifications. The underlying philosophy in this approach is one that assumes elimination of a large number of possible solutions with increasing detail in specifications. This hierarchical decomposition is described in terms of the portal frame problem as follows.

A design with all previous specifications and an additional requirement of minimum weight must be obtained. While all conditions cannot be transferred to the component level, the structural generation problem can be viewed as designing each component separately for whatever specifications are applicable. Beam element A is sized for each of the applicable specifications. Its length is determined by distance from load P_1 to the support point S_1 . Similarly, the cross sectional type and the corresponding section dimensions are obtained from strength and local buckling considerations. At the component level, however, sufficient detail is not available to see if component level design satisfies the specification of the global structural stiffness. A recommended procedure at this level is to determine the sensitivity of global structural stiffness to local component variables, and to use this information when an assemblage of the components is done.

In addition to processors that implement strategy, testing operators comprise an important component of the function level. The controller is faced with the formidable task of directing execution of generators and testers in an efficient sequence. Clearly, if a generated concept fails an acceptability test, several remedial measures are available for implementation. The simplest entails a backtracking to the last decision, and revising that decision with the failure as a constraint. As an example, if a square cross section was selected for a beam element to satisfy stiffness and strength requirements and was later found unfit from a manufacturability standpoint, one would simply backtrack to the decision of choosing a cross section with the additional manufacturability constraint. Another frequently used approach, and described by Tong [13] as "pruning", is especially applicable if specification decomposition allows construction of tree-like deduction paths. Here, failure of a

partial design can result in eliminating several possibilities from the search space.

Finally, the controller must have the option of modifying the design rules, particularly if it assists in realizing the design specifications. The acceptability tests can themselves be relaxed to admit designs. Examples of this include relaxation of manufacturability requirements to admit non circular cross sections. Likewise, allowable values of stress or displacements can be modified to pass the acceptability test. This concept is particularly powerful, if critical satisfaction of constraints in partial designs is consciously avoided. Yet another option available at this stage is to extend the design without replacing the current design. This translates into adding features which work with an existing design to enable it to pass the acceptability requirements.

Program Level

The foregoing discussion details the requirements and assigned tasks of the knowledge and function level. The mechanics of implementing all the design steps, including programming procedures, production rules, and database management systems is relegated to the program level. No problem solving knowledge is available at this level - it simply implements and manages instructions passed in from the other levels.

Particular attention must be directed at the database management capabilities of such a system. Significant amounts of data are generated and must be managed for a design system to work efficiently. This is even more crucial as large amounts of algorithmically generated numerical data must be stored and post-processed to use meaningfully in the iterative process. Two levels of data management are planned in the current system. The global data base is at the core of such a system and records information for long term usage. Problem and subproblem related databases are extracted from the main system and are local to the knowledge level. This provides a convenient blackboard for constraint posting and propagation as the design is taken through a process of incremental refinement (Figure 5).

The inference facility is another important feature at this level. In the structural design problem that is currently under implementation, a rule-based, C Language Integrated Production Systems (CLIPS) [21] is being used. This utility can be invoked from within a FORTRAN program, making available a convenient link between algorithmic and heuristic processing of information.

Optimal Topology Assignment

The basic goals of an optimal topology generation system within the framework of a problem solving system described in previous sections is outlined here for completeness. A set of load conditions and support points are defined in a design space.

The design space also consists of obstacles and prohibited zones in which no portion of a structural assembly may be placed. An optimal, minimum weight structural topology is to be generated to transfer the applied loads to the supports, satisfying requirements of allowable stress in structural members, displacements at load points, and limits on component and system static stability.

The types of structural elements that may be used in the structural synthesis are limited to axial force members (tension only, and tension/compression), flexural beam elements, and membrane elements (triangular and quadrilateral). In addition to these primitive elements, assembly of axial force elements (tension/compression) in a triangular truss is also available as a master element.

The topology generator is first invoked to construct a series of structural assemblies that stabilize the applied loads. This is an incremental process which attempts to meet the problem specifications in one of two ways. The first approach looks at each load sequentially, assessing its geometric orientation with respect to the supports, and selects an appropriate element to provide partial stability. At any step of the generation, a branching can be introduced to implement more than one acceptable alternative. A second approach divides the structural domain into four quadrants, and a structure is generated in each quadrant to account for loads in that region. The substructures are then connected by acceptable least weight elements. At this stage of the problem, the only active problem specifications are those related to the geometry of the load distribution and the applicable element types.

A sequence of refinements is made to these configurational possibilities, with each step accounting for one design specification from an ordered list. Such an approach assumes that a set of alternate designs optimized in this manner are better suited to identifying the most promising configuration for detailed design. A second approach that is planned for implementation in the proposed study uses the topology generator to seed the design space with possible alternatives. An optimal topology is then obtained by a combination of the most favorable characteristics of the seed designs. The generate and test approach outlined above relies to a large extent on algorithmic processing and efficient handling of numerical data. The three tier organization of the problem solving system described in preceding sections is ideally suited to this complex task. Additional details of the implementation will be presented in [22].

Closing Remarks

The present paper presents the framework of a knowledge-based system for structural design. The process of design of a structural system includes the initial structural geometry

definition followed by successive refinement of this initial configuration to obtain an optimum. The system described for this task is distinct from previous systems in this domain. The latter were largely restricted to consultative tasks. The use of decomposition principles to make the design problem more tractable is very similar to multilevel decomposition techniques proposed in automated optimum synthesis of structures. A formalism in the organization of such systems is considered very important if significant advances in problem solving capabilities are to be realized. Finally, the role of integrating algorithmic and heuristic processing of databases is considered vital for the success of such systems.

Acknowledgments

The author gratefully acknowledges his discussions with doctoral student N. Sangameshwaran during the preparation of this paper.

References

1. J.L. Chen and P. Hajela, "FEMOD: A Consultative Expert System for Finite Element Modeling", proceedings of the AIAA/ASME/ASCE/AHS SDM Conference, April 6-8, 1987, Monterey, California. Computers and Structures, Vol. 29, No. 1, 1988.
2. J.L. Chen and P. Hajela, "A Rule Based Approach to Optimum Design Modeling", Computer Applications in Structural Engineering, (ed. D.R. Jenkins) ASCE, New York, pp. 66-81, 1987.
3. P. Hajela and J.L. Chen, "CAD Interfaces in the Development of Expert Systems for Optimum Structural Design", Proceedings of the Second International Conference in Computational Engineering Science, Atlanta, Georgia, April 1988, Computational Mechanics, Springer Verlag, April 1988.
4. A. Hart, "Knowledge Elicitation: Issues and Methods", Computer Aided Design, Vol. 17, No. 9, November 1985, pp. 455-462.
5. J. Bennett, L. Creart, R. Englemore and R. Melosh, "SACON: A Knowledge Based Consultant for Structural Analysis", Technical Report STAN-CS-78-699, Stanford University, Stanford, California, 1978.
6. S.C.-Y. Liu, "A Consultative Expert System for Finite Element Modeling of Strip Drawing", presented at XIII North American Research Conference, University of California at Berkeley, May 1985.
7. R.H. Holt and U.V.L. Narayana, "Adding Intelligence to Finite Element Modeling", Expert Systems in Government

Symposium (eds. K.N. Karna, K. Parsaye and B.G. Silverman), IEEE, New York, 1986, pp. 326-337.

8. J.J. Shah, "Development of a Knowledge Base for an Expert System for Design of Structural Parts", Proceedings of the ASME International Computers in Engineering Conference, 1985.
9. J.J. Shah, "Deziner - An Expert System for Conceptual Form Design of Structural Parts", Proceedings of the ASME International Computers in Engineering Conference, 1986.
10. G.E. Nevill, L.A. Jackson, and J.H. Clinton, "Automated Hierarchical Planning for Structural Design", proceedings of the ASME International Computers in Engineering Conference, San Francisco, California, August 1988.
11. G.E. Nevill, and G.H. Paul, "Knowledge Based Spatial Reasoning for Designing Structural Configurations", proceedings of the 1987 ASME International Computers in Engineering Conference, New York, Vol. 1, pp. 155-160.
12. J.P. Brown, "Managing Interacting Design Subproblems in Incremental Preliminary Design", M.S. Thesis, Department of Aerospace Engineering, Mechanics and Engineering Science, University of Florida, August 1988.
13. C. Tong, "Toward an Engineering Science of Knowledge-Based Design", Artificial Intelligence in Engineering, Vol. 2, No. 3, 1987, pp. 133-166.
14. C. Tong, "KBSDE: An Environment for Developing Knowledge-Based Design Tools", proceedings of Knowledge Compilation Workshop, Oregon State University, September 1986.
15. D. Whetstone, "SPAR - Reference Manual", NASA CR-145098-1, February 1977.
16. P. Hajela, "Genetic Search - An Approach to the Nonconvex Optimization Problem", in review for presentation at the 30th AIAA/ASME/ASCE/AHS/ASC SDM Conference, Mobile, Alabama, April 1989.
17. G.N. Vanderplaats, "An Efficient Feasible Direction Algorithm for Design Synthesis", AIAA Journal, Vol. 22, No. 11, October 1984, pp. 1633-1640.
18. J. McCarthy and P. Hayes, "Some Philosophical Problems from the Standpoint of Artificial Intelligence", Readings In Artificial Intelligence, Nilsson, 1981.
19. J.E. Sobieski, A Linear Decomposition Method for Larger Optimization Problems - Blueprint for Development, NASA TM-83248, 1982.

20. J.E. Sobieski, B.B. James, and A. Dovi, "Structural Optimization by Multilevel Decomposition", AIAA Journal, Vol., 23, No. 11, 1983, pp.1775-1782.
21. CLIPS Reference Manual, JSC-22948, NASA Johnson Space Center, April 1988.
22. P. Hajela and N. Sangameshwaran, "A Coupled Algorithmic and Heuristic Approach for Optimum Structural Topology Generation", in review for International Conference in Computer Aided Optimum Design of Structures, 20-23 June 1989, Southampton, England.

C-4

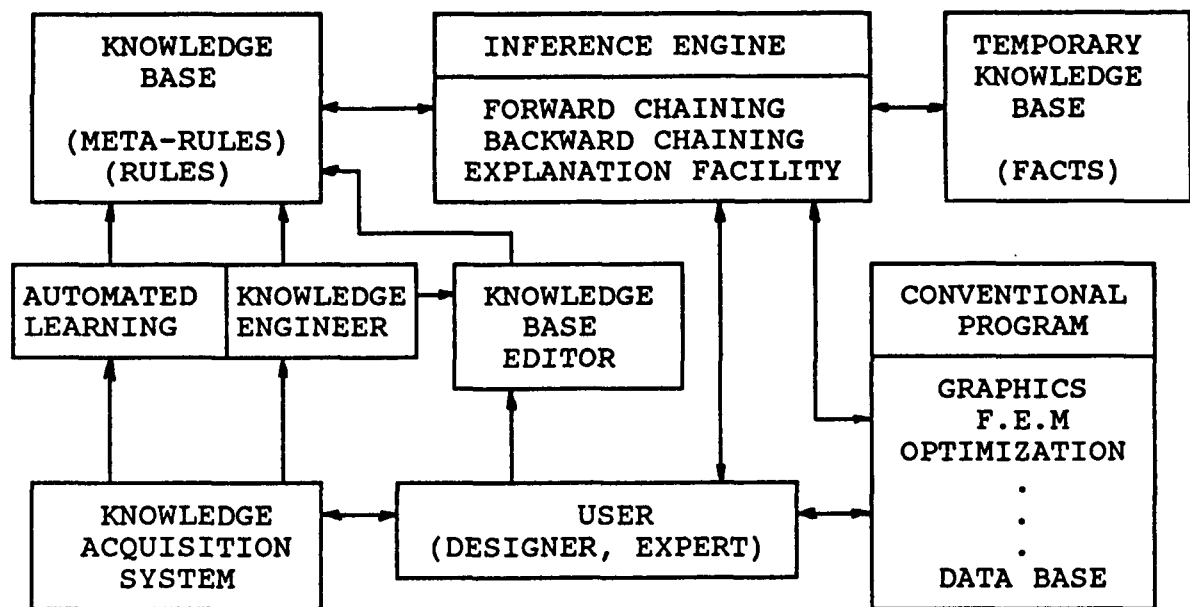


Figure 1. Architecture of the OPSYN system

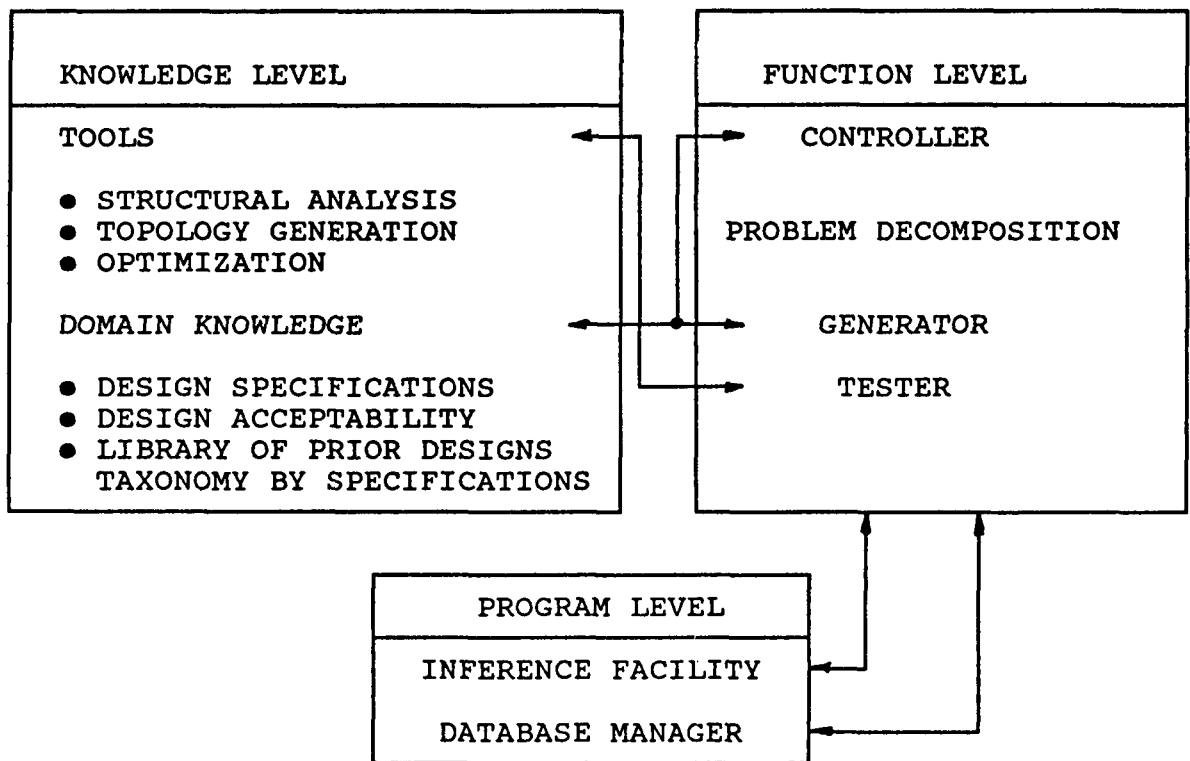


Figure 2. Organization/Subdivision of tasks in a typical problem solving system.

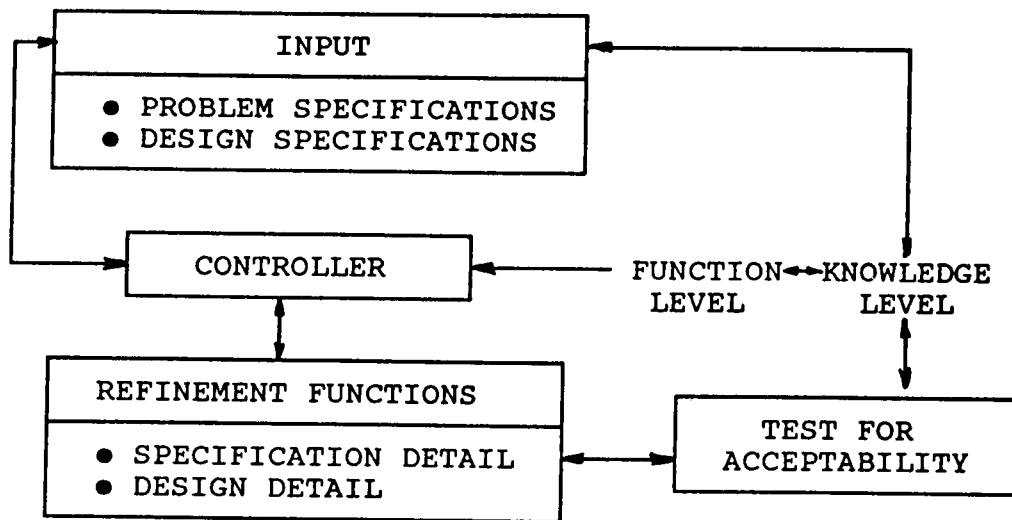


Figure 3. Schematic of a refinement operator

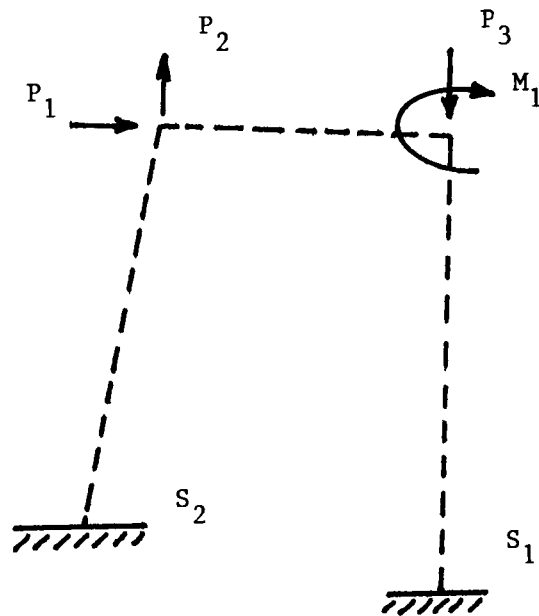


Figure 4. The portal framework problem.

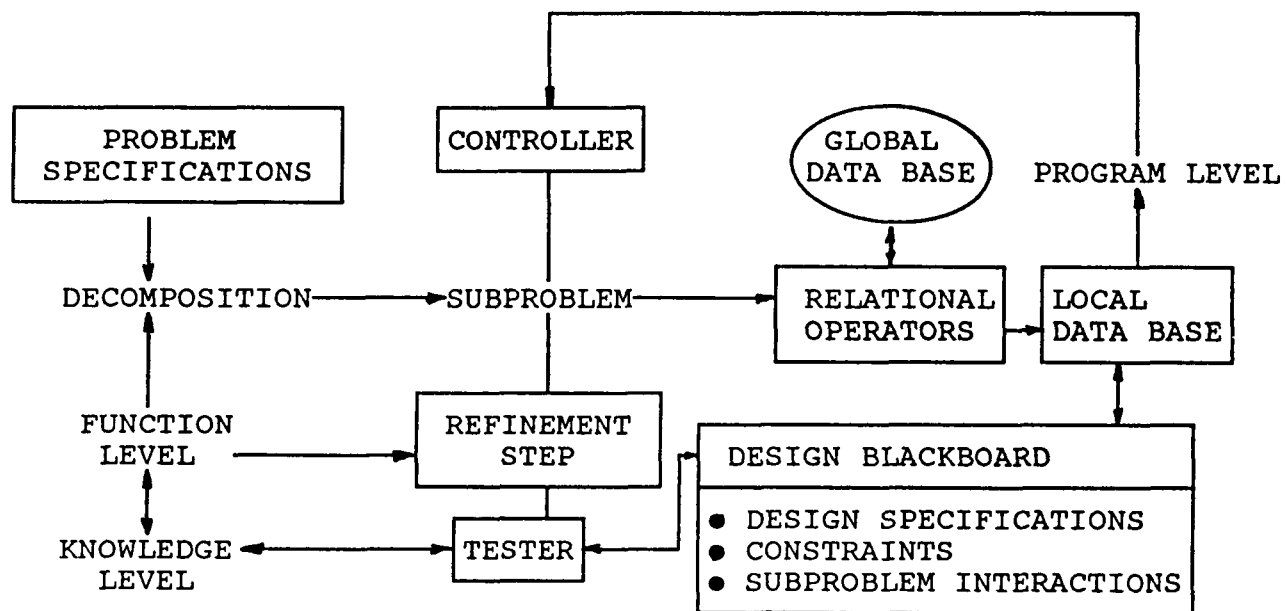


Figure 5. Significance of database management in problem solving systems.