A Hybrid Reasoning System for Conceptual Design

B.D. Netten

Delft University of Technology
Faculty of Information Technology and Systems
Zuidplantsoen 4, 2628 BZ
Delft, The Netherlands
B.D.Netten@cs.tudelft.nl

Abstract

A hybrid system is presented that supports conceptual design as a progressive reasoning process. Alternative solutions are generated and evaluated at several levels of abstraction to find the best concept. The system consists of four components that are applied sequentially. This approach avoids integration problems between the system components. This approach is applied in EADOCS, a design system for composite sandwich panels.

Introduction

Conceptual design is the first phase in a design process. The objectives for a conceptual design study are to search for the best concept, evaluate its feasibility and decide on a go-ahead for preliminary design.

Here, conceptual design is regarded as an innovative design process (Brown 1991). Concepts are designed from predefined types or abstractions of solutions. Usually, it is unknown what the globally optimal solution is. Finding the best concept then involves extensive and explorative search through the design spaces of many types of solutions.

Conceptual design is an iterative search process at various levels of abstraction through the design spaces of solution types. Reasoning about the feasibility and global optimality of solution types is initially at a high level of abstraction. The level of abstraction is reduced when design reasoning zooms in to refine solution. New design parameters and constraints are introduced when entering the design space of a solution type.

While exploring different types of solutions, a designer may discover new opportunities or problems and elaborate the initial specifications.

Supporting design reasoning at different levels of abstraction, with different sources and representations of domain knowledge, requires a hybrid system in which the system components have a specific task in the design process. When several reasoning components are integrated, several problems may be introduced:

- Knowledge bases are redundant or inconsistent.
- Transformations between knowledge representations are ambiguous.
- Conclusions are ambiguous or inconsistent.

The roles assigned to components determine the significance of redundancy and ambiguity. For example, when system components are competing to hypothesize conclusions, their ambiguous or inconsistent solutions cannot be compared. When the design process iterates between several components, ambiguous or inconsistent results cannot be transformed into other representations.

Similar problems on ambiguity and inconsistency have already been identified for manual design. Pugh (1981) proposed a progressive approach to conceptual design reasoning in which ambiguities and transformations are minimized. This approach can also be applied for a hybrid system.

This paper presents how Pugh's approach is applied in EADOCS, a system for Expert Assisted Design Of Composite Sandwich panels.

The next sections characterize the design approach, application domain, EADOCS' system components and their integration. An example of a design session shows how each component generates intermediate solutions. This paper only presents the tasks of the system components in the reasoning process. More information on the implementation or application domain can be found in (Netten 1997, Netten and Vingerhoeds 1997).

Conceptual design approach

Conceptual design is an explorative search to find the best concept, in which many alternative solutions are generated, evaluated, compared and refined. Designs are generated in a progressive approach.

Initially, several solutions are generated, evaluated and compared at a high level of abstraction. The best solutions are selected as the set of alternatives for refinement in following iterations.

In one iteration, solutions are generated and evaluated at the same level of abstraction to minimize ambiguity in the comparison and selection. The level of abstraction is gradually refined in successive iterations.

Each alternative is a starting point for a new iteration. The new design problem is elaborated for the alternative solution of the starting point.

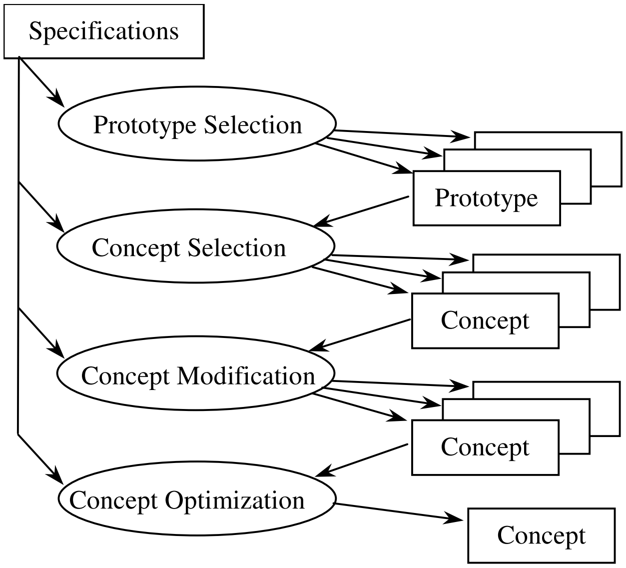

Figure 1: Progressive design approach

This progressive design approach has several significant advantages over a greedy approach, such as:

- Ambiguity in the evaluation and comparison of alternative concepts is reduced.
- A more global approach to iterative search is provided in which many different solutions are evaluated.
- The iterative design process is decomposed into several smaller and more manageable iterations.
- Efforts on exploring details are reduced or postponed to following iterations.

A progressive approach also provides potential advantages for the development of a hybrid system, such as:

- The iterative design process can be decomposed in a sequence of design phases with smaller iterative loops.
- In each phase, designs are refined at a particular level of abstraction.
- Solutions and design problems are only refined and not transformed back to the representation of previous design phases.
- Multiple reasoning components can be developed for each design phase, with a different knowledge base, representation, and inference engine.

Design abstraction

A conceptual design defines the types of components, their composition, and values for their primary design parameters. Most of these parameters have qualitative or discrete values. Secondary design parameters for joints or other design details are only introduced for preliminary or detailed design and not considered here.

At least two levels of abstraction can be identified for conceptual designs. At the highest level, designs are represented as prototypes (Rosenman and Gero 1993), defining the types of components and their composition.

At a lower level, components and their compositions are represented as objects with their parameters and relations. These solutions are called concept solutions, or concepts, and are instantiations of prototypes.

Design phases

The progressive design approach distinguishes several successive design phases for the generation and refinement of prototypes and concepts (Figure 1). Each phase is an iterative process on a specific level of abstraction. Four design phases are identified:

1. Prototype Selection (PS)
2. Concept Selection (CS)
3. Concept Modification (CM)
4. Concept Optimization (CO)

In the first phase, prototype solutions are generated by selection of types of components, materials and configurations, and by composition of the selected solutions. Each prototype will be instantiated and refined individually in the following phases.

In the second phase, concept solutions are generated for a prototype solution. A concept is generated by retrieval of the solutions from previous design cases. A selected concept is heuristically modified and numerically optimized in the third and fourth design phases respectively.

Design process

Solutions can be represented hierarchically as nodes in a tree. A node branches to refined solutions generated in a next iteration. Pugh proposed to expand the tree in a breadth first search, but a designer may also apply other strategies.

While designing solutions, new constraints and preferences for solutions may be identified and added to the problem specification. This elaboration of specifications is not explicitly drawn as arrows in Figure 1. Usually, an elaboration concerns only the design that is currently refined and will not affect any of the previous solutions.

An elaboration of specifications may also be more drastic and affect previous iterations. Feasibility and optimality of previous solutions has to be re-evaluated. The ordering of alternative solutions might also be affected and, therefore, the previous design phases should be re-examined. As indicated in Figure 1, specification changes are made at the top of the process and restart design from phase 1. Whether a design phase is completely repeated or only extended, depends on the implementation of the modules.

It should be noted that the design objective is to evaluate and compare all relevant concepts. Ambiguities in the evaluation of solutions should be avoided. Backward tracing of effects on individual designs, or an iterative approach in which detailed results are fed back to previous design phases, are therefore not desired.

Application domain

Two types of applications have been considered during the development of EADOCS.

- EADOCS supports the conceptual design of composite sandwich panel structures, and is a pilot application for development of the reasoning system.
- AIDA project (Artificial Intelligent Design of Aircraft, Rentema et al. 1997) supports the conceptual design of civil aircraft and uses part of the EADOCS kernel.

Although the scopes of these applications are different, they share several design characteristics.

Both applications address an innovative reasoning process in which new solutions are obtained in a search through a predefined set of solution spaces for prototypes. The most difficult problem is the search for optimal structures and components; i.e. selecting, instantiating and composing prototypes. This search is supported by a combination of search techniques and knowledge sources.

Sandwich panels

Panels are structural components applied in most aircraft. A panel consists of a skin and some type of stiffening to maintain structural stability. One type of stiffening is to sandwich a core of honeycomb material between the skin and an extra inner-skin. The skin can be made as a laminate of one or more layers of fiber reinforced composite material. Each layer consists of many plies of the same material and with the same fiber orientation. The laminate lay-up of a skin defines the order and orientation of the layers. For example, a symmetrical laminate with a ±45° cross-ply outer layer and a 0° uni-directional inner layer is denoted as $[\pm 45/0]_s$.

The type of stiffening, materials, and laminate lay-up, determine the behavior of a panel, such as strength, stiffness, weight and cost. The objective is to design a panel that is feasible for the constraints on strength and stiffness, and that is optimal for weight or cost.

The problem can be defined as a numerical optimization problem with a linear optimality criterion, discontinuous and non-linear constraints, and discrete variables. Numerical routines exist for the analysis of a specific type of panel and structural stability phenomenon. The objective for conceptual design is to select the best panel configuration and values for the discrete variables.

EADOCS components

EADOCS has four reasoning components, one for each design phase. The components are applied sequentially and interact only via intermediate solutions. A component receives input from the previous phase for the elaborated specifications and an initial solution, and provides one or more refined solutions as output to the next phase. EADOCS has a common object oriented data structure to represent problems and solutions. The data structure consists of three substructures for the structural, functional and behavioral models. Data for components is only transformed to and from this common data structure to reduce the integration and development efforts of system modules. The reasoning components are presented in the order of the design process.

Prototype Selection

The prototype selection phase is implemented as a constraint-based reasoning component. Design variables are defined for the types of panels, components, materials, and laminate lay-ups. The behavior and functionality of variables is represented qualitatively as relative grades of performance. These grades are acquired from literature on domain theory and experimental results.

Constraints are defined for the behavioral and functional requirements and for combination of variables into prototypes. Constraints are defined as thresholds on qualitative performance of variables.

First, the specifications are qualified into required performance grades for the constraints. Constraints are satisfied by selecting solutions that meet the required grades. Solutions are propagated to other constraints by the order of their performance grades. Prototypes are generated from combinations of individual solutions that satisfy all constraints. Feasibility and optimality of prototypes are evaluated from their aggregated grades of performance.

The initial specifications can be elaborated in two ways.

- When no feasible prototypes can be found, one or more requirements can be relaxed by reduction of the required performance grades. Likewise, requirements can be raised to reduce the number of prototypes.
- The network of constraints also activates functional and behavioral relations that are not specified initially. From these constraints, preferences can be selected for prototypes, and additional requirements can be specified.

Concept Selection

The concept selection phase is implemented as a case-based retrieval and reuse component. As a post-processing step to the design process, it also retains optimized design cases. The case-base contains a relatively small number of designs, including their structure, functionality, and behavior.

Cases are indexed in separate memory structures for the design structure, functionality, and behavior. Cases are indexed by their prototype solution, types of components, and classified by their functionality and behavior onto the discrete values applied during prototype selection.

An initial target for retrieval is defined from the elaborated specifications and a prototype solution. Initially, the complete structure of a case is retrieved. It is up to the designer which and how many cases to retrieve and reuse.

Initially, cases can only be retrieved for intended functionality and behavior. Feasibility for other requirements should either be retrieved from cases with a similar structure, or analyzed numerically. In the reuse phase, cases can also be adapted for remaining specifications, by combination with parts from other cases (Netten and Vingerhoeds 1997b). This serves three design objectives:

- When a case has not been design for a specified requirement, the feasibility of its solution cannot be retrieved. Retrieval of cases with a similar structure could provide a prediction on feasibility.
- When the retrieved solution is infeasible for a specified requirement, a repair can be retrieved from a case with a similar structure too.
- To retrieve more drastic structural adaptations that cannot be made during revision, such as the modification of laminate lay-ups.

Concept Modification and Optimization

A selected concept is revised in two steps; heuristic modification and numerical optimization. First, the symbolic and discrete parameters are modified heuristically. This is implemented in a rule-based reasoning component. Heuristic rules are defined from domain theory for several well-known repairs and improvements. The heuristics suggest discrete and local adaptations of layer materials, orientations and thickness. Some adaptations for discrete variables such as materials cannot be suggested from numerical optimization routines. These values have to be improved before numerical optimization is useful. Other adaptations suggest discrete steps for numerical parameters as shortcuts to reduce numerical search in the fourth step. These parameters are locally optimized in the fourth design phase with Box's Complex method (see also Van Bladel 1995).

Integration of CBR in the design system

The case-based reasoning component performs two essential design tasks by instantiating concepts for prototype solutions and by adapting the structure of concepts. These two tasks cannot be performed by any of the other components, because the domain knowledge is not available. Heuristic or numerical operations to adapt laminate lay-ups are inaccurate not unavailable.

The other components, however, also perform essential tasks that could not be performed efficiently by case-based reasoning. In this application, domain knowledge is available in the form of constraints, cases, rules and analysis models. The knowledge is also represented in these forms to minimize the development efforts.

In the development of EADOCS and AIDA, it was assumed that the number of design cases available was relatively small. The coverage of the case-based reasoning module is therefore also restricted, even though case combination could extend its capabilities.

A major disadvantage of a small case-base is that the feasibility and optimality of prototype solutions cannot be retrieved or compared. Additional domain knowledge, in the form of performance constraints is necessary for prototype selection.

Another disadvantage of a small case-base is the necessity for accurate revision to complement the structural adaptations by reuse of case components.

Experimental results have shown that although each component has a different knowledge base, inconsistencies in suggested solutions do not pose significant problems for design support. Each system component is assumed to be more accurate and detailed than its preceding components, and its conclusions overrule previous conclusions.

Each component has a well-defined and complementary task. If a component cannot provide adequate support, for example when domain knowledge is not available, this cannot always be compensated in other components. Consequently, the best concept is not always the global optimum.

Example of a design session

The following example shows how each of the system components performs its subtask in the conceptual design process and how it generates or refines solutions. It should be noted that for clarity many design steps and results have been left out.

The initial problem specifies requirements for the dimensions of the panel (1500 x 1200 mm^2, or 4.6 x 3.7 ft^2), and a set of seven different loading conditions. Each loading condition is a combination of in-plane tension, compression and shear loads. All loads are less or equal to 3000 N/mm, or 5 lb_f/inch. The panel should be optimized for minimum weight.

Prototype selection

The first task is to generate prototype solutions and evaluate the initial specifications. This is achieved by constraint satisfaction.

The first step is to qualify the constraints for loading conditions and panel dimensions:

- The dimensions are qualified as a large panel.
- The loads are qualified as light or medium loads.
- The panel stability (a combination of dimensions and in-plane loads) is qualified as a medium-buckling load.

This example defines a problem where the stability of the panel will be a critical constraint.

The second step is to select prototype solutions that best satisfy one or more of the constraints. Prototype solutions are abstractions of conceptual solutions. Prototypes are defined for the types of panel, classes of material, layer types and orientations. Table 1 gives the prototype solutions selected for this session:

- The panel types are selected for their specific stiffness to medium buckling loads.
- Combinations of in-plane tension and compression require layers of uni-directional fibers in all directions, while shear requires cross-ply layers.
- Optimality for minimum weight requires materials with high specific strength and stiffness, such as the carbon and aramid fiber reinforced plastics (cfrp, afrp).

The third step is to propagate the individual solutions to combine prototype solutions for a complete panel. A sandwich panel for example can be combined with any of the layer and material prototypes.

Finally, the initial specifications are elaborated with preferences for prototype solutions. Each combined prototype is regarded as an alternative elaboration of the initial specifications, and is treated separately for concept selection. This example only elaborates on the sandwich panel prototype.

Concept selection

The second phase selects concept solutions for an elaborated problem specification. Concepts are selected by reusing case solutions.

In the first step, complete case solutions are reused. This step is identified as design phase 2a. The initial target retrieves solutions that are similar to the initial specifications and the preferred prototype solution.

The combination of the seven loading conditions has not been solved by any of the cases. The five most similar cases match only two loading conditions. Three other cases match also a third loading condition, but their dimensions are much smaller than required. These eight cases can be grouped in four different laminate lay-ups (Table 2, Retrieved lay-up). All eight cases are reused as alternative concepts and require adaptation for the remaining loading conditions and dimensions.

Before revision of the precedent solutions in phase 3, the case-base is consulted again for integrating parts from other cases. For this second step (design phase 2b), the target set to retrieve parts from cases that are similar to the precedent case also satisfy one or more of the remaining targets.

Most important are adaptations of laminate lay-ups that cannot be suggested from specialist operations in phases 3 or 4. Table 2 (Combined lay-up) shows that the laminate lay-ups of cases in groups I and II could be adapted by insertion of other essential layer types.

The primary objective in design phase 2b is to obtain a feasible design. Therefore, only conservative adaptations are reused, which are usually unfavorable for optimality. Conservative adaptations are for example the addition of material, or the insertion of a layer in a laminate.

Figure 2 shows the convergence of the alternative solutions during the design process. The adaptations in 2b increase the panel mass to restore feasibility for remaining targets.

Table 1: Prototype solutions

	Class of solutions
Stiffening	Sandwich panel Hat-stiffened panel
Layers	Uni-directional in 0 and 90 degrees Cross-ply material in ±45° or in (0-90)°
Material	CFRP AFRP

Table 2: Retrieved laminate lay-up structures

	Retrieved lay-up (2a)	Combined lay-up (2b)
I	$[\pm45/0]_s$	$[\pm45/0/90/HC]_s$
II	$[0/\pm45/HC]_s$	$[0/90/\pm45/HC]_s$
III	$[0/90/0\text{-}90/HC]_s$	
IV	$[\pm45/0/90/\pm45/HC]_s$	

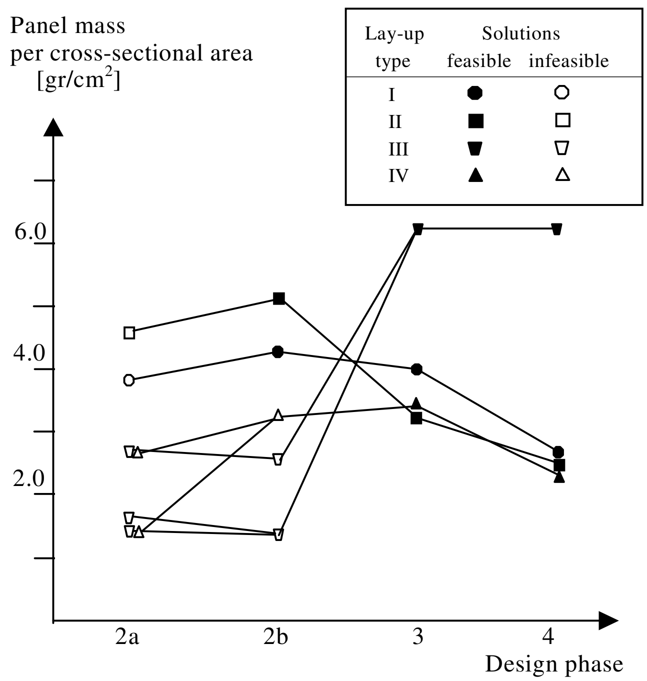

Figure 2: Optimization of panel mass

Concept Modification

The third design phase heuristically modifies a concept to improve feasibility and optimality. These modifications, or revisions, require the results from analysis routines as critique on behavior. The heuristic rules are specialist operations to repair or improve discrete layer properties for the critique on behavior.

Figure 2 show the effect of heuristic modification. The infeasible concepts from phase 2b are repaired, and the optimality of the feasible concepts has improved. Many of the heuristic modifications could also have been suggested in phase 2b, if only the appropriate cases would have been available. Usually, the heuristic modifications are proportional to panel weight. More drastic adaptations of panel type, laminate lay-up, or materials, have a smaller effect on weight, but are too complex for specialist operations. These design decisions are assumed to be optimal for a concept resulting from phase 2b.

Concept Optimization

In a final phase, the layer thicknesses of a modified concept from phase 3 are numerically optimized. The layer thickness is a discrete optimization variable for the number of plies. Box's Complex method is applied (Van Bladel 1995). Figure 2 clearly shows the effect of the laminate lay-up on optimality. Lay-up type IV results in the best concept, which is 2.5 % lighter than for type II.

Conclusions

A hybrid system is presented in which four components are applied sequentially to support a progressive approach to conceptual design. Each component supports a specific design phase and interacts to other components by passing intermediate results via a common data structure. Integration problems, such as redundancy, ambiguity and inconsistencies, are minimized and do not pose significant problems in design reasoning.

References

Brown, D.C. 1991, Routiness Revisited. In *Mechanical Design: Theory and Methodology*, Waldron M. and Waldron K. eds., Springer Verlag.

Maher, M.L., Pu, P., 1997, *Issues and Applications of Case-Based Reasoning in Design*. New Jersey: Lawrence Erlbaum.

Netten, B.D. 1997. Knowledge Based Conceptual Design: An Application to Fiber Reinforced Composite Sandwich Panels, Ph.D. thesis, Delft University of Technology.

Netten, B.D., and Vingerhoeds, R.A. 1997a. EADOCS: Conceptual design approach in Three steps. *Engineering Applications of Artificial Intelligence* 11:341-342.

Netten, B.D., and Vingerhoeds, R.A. 1997b. Structural Adaptation by Case Combination in EADOCS. In 5th German Workshop on Case-Based Reasoning, 171-180. Bad-Honnef, Germany, LSA-97-01E, Dept. of Computer Science, Univ. of Kaiserslautern.

Pugh, S. 1981. Concept Selection – A method that works. In Int. Conf. on Engineering Design ICED81, Hubka V. ed.: 497-506.

Rentema, D.W.E., Jansen, F.W., Netten, B.D., Vingerhoeds, R.A., 1997. An AI-Based Support Tool for the Conceptual Design of Aircraft. In Computer Aided Conceptual Design '97, Bradshaw, A., and Counsell J. eds.: 47-55.

Rosenman, M.A. and Gero, J.S., 1993. Creativity in Design Using a Design Prototype Approach. In Modeling Creativity and Knowledge-Based Creative Design, Gero, J.S. and Maher, M.L. eds.: Lawrence Erlbaum Ass., Chapter 6, 111-138.

Van Bladel, P.G. 1995. Design of Fibre Reinforced Composite Panels for Aerospace Applications, Ph.D. thesis, Delft University of Technology.

Appendix

1. Integration name/category EADOCS

2. Performance Task

Conceptual design (structural design, and parametric design of discrete and primary variables)

3. Integration Objective

Progressive design support: design support at increasing levels of detail and accuracy.

4. Reasoning Components

1. Constraint-based reasoning in Prototype Selection component (PS).
2. Case-based reasoning in Concept Selection component (CS).
3. Rule-based reasoning in Concept Modification component (CM).
4. Numerical optimization in Concept Optimization component (CO).

5. Control Architecture

Sequential, in the order mentioned in 4:

PS → CS → CM → CO

6. CBRCycle Step(s) Supported

Pre-processing in PS
Retrieval, reuse and retention in CS
Revision in CM and CO

7. Representations

Qualitative behavioral constraints in PS
Design cases in CS
Heuristic modification rules in CM
Numerical analysis models in CM, CO

8. Additional Components

Numerical optimization routine

9. Integration Status

Emperical evaluation:

- comparison of suggestion for solutions with opinion of domain expert
- comparison of designs with numerical optimization results

10. Priority future work

- Extension of domain knowledge for other types of structures and modifications.
- Learning of refinements of qualitative behavioral constraints and heuristic modifications.