Structural Adaptation by Case Combination in EADOCS

B.D. Netten[1], **R.A. Vingerhoeds**[1,2]

[1)] Delft University of Technology, Faculty of Technical Mathematics and Informatics, Zuidplantsoen 4, 2628 BZ Delft, Netherlands, [bart | rob]@kgs.twi.tudelft.nl

[2)] University of Wales Swansea, Dept. of Electrical and Electronic Engineering, Singleton Park, Swansea SA2 8PP, United Kingdom

Abstract

EADOCS (Expert Assisted Design Of Composite Sandwich panels) is a multi-level and hybrid expert system for innovative conceptual design. The objectives for conceptual design support are to assess the feasibility and optimality of alternative solutions and to find the best concept for further detailed design. Case-based reasoning is applied to generate conceptual designs and to refine their structure. Specialist operations for structural adaptation are not available in EADOCS and structural adaptations have to be retrieved from the case-base. This paper presents a domain independent approach to retrieve and reuse case components for structural adaptation.

1 Introduction

The design cycle can be divided in three phases; conceptual, preliminary and detailed design. The conceptual design phase is the first phase starting with an initial specification of design requirements and preferences for the solution. The objective is to find the "best" design, which involves the configuration of a design structure and the assignment of design parameters. The complexity of the design problem requires an incremental design approach in which the configuration and parameter values are gradually established. Cost and time for designing strongly increase during the design cycle. Conceptual design involves abstract design of the configuration and primary parameters. Adaptations of this concept during later phases are very expensive and time consuming. The efficiency of the design process, measured in design time, effort and the optimality of the final design, strongly depends on the number, the feasibility and optimality of conceptual designs. The tasks for conceptual design are to evaluate the feasibility and optimality of alternative configurations and to generate a minimal number of near-optimal concepts as starting points for following design.

EADOCS is developed to support the conceptual design of sandwich panels [Netten '97]. EADOCS implements a progressive reasoning approach to support the generation, adaptation and evaluation of conceptual designs (see section 3.2). This is an innovative design process[*] in which a panel has to be configured from a set of predefined layouts, components and materials. There is no single design procedure to determine the optimal configuration for any set of design specifications. The structural and functional models for panels are predefined from domain knowledge (see section 3.1). Detailed routines are defined for the behavioural analysis of panel layouts.

The behaviour of a component strongly depends on the design structure and the behaviour of other components. The behaviour and functionality of a panel design is therefore determined by its configuration. The strong component interaction is characteristic for design problems addressed in EADOCS:

[*] Classes of design are defined in [Brown and Chandrasekaran '89].

© B.D. Netten, R.A. Vingerhoeds, 1997

- Accurate evaluation of specified requirements on design behaviour and functionality is only possible for a completely defined concept.
- Functionality and behaviour of a panel cannot be directly related to a single component, a dimension or a material.
- Many different configurations may satisfy design requirements after parametric optimisation. Their values for the optimality criteria, however, may vary significantly. The configuration is therefore crucial for optimality and should be selected upon before parametric design and optimisation can commence.

Case-based reasoning plays an essential role in EADOCS for the generation of initial concepts. The number of available cases, however, is too small to provide a good coverage of the design space and to predict feasibility and optimality of configurations and components. Additional design phases are necessary before and after the retrieval and reuse of cases to evaluate optimality of configurations and to adapt concepts. Parametric adaptation is performed in two steps of heuristic modification and numerical optimisation. For structural adaptation, however, two problems arise:

- Structural adaptation is complex due to the strong component interaction. Heuristics or theoretical models are either unavailable or very difficult to acquire.
- The case-base may not contain designs for an optimal configuration.

Some considerations are made for case-based reasoning in the application domain of EADOCS in the sections 2 and 3. In the absence of appropriate domain knowledge for structural adaptation, EADOCS implements a domain independent approach for the combination of cases (see sections 4 and 5).

2 Some considerations on case-based reasoning for conceptual design

Case-based reasoning can be an efficient technique for design support (e.g. [Maher et al. '95]). Design applications are complex and operations to generate a (sub)solution for design requirements are usually not available. Case-based reasoning provides a mechanism to generate initial designs. In many applications, the design structure provides also a structure for case definition, similarity assessment, retrieval and reuse (e.g. [Börner et al. '96]). EADOCS has a hierarchical data structure in which classes are defined for the design structure, functionality and behaviour. Cases are also defined hierarchically by objects for its structural solution, functionality, and analysis results for behaviour. The evaluation and adaptation operations are defined for each class of structural solutions and are not defined within each case.

In EADOCS, as in many other design applications, design problems cannot be decomposed into tractable subproblems. All components contribute to some extend to each of the individual design requirements. An individual design requirement is a subproblem, and the combination of design requirements is characteristic for the concept solution. Solutions to subproblems cannot be expressed as parts or snippets of previous design cases, and behaviour of components cannot be abstracted to predict overall design behaviour as for example in CADET [Sycara et al. '92]. Subproblems cannot be indexed for the retrieval of particular structural solutions, as for example in JULIA [Hinrichs and Kolodner '91], Déjà Vu [Smyth and Keane '95], or PARIS [Bergmann and Wilke '96] (discussion in [Netten and Vingerhoeds '96]). Dynamic problem decomposition to solve a subproblem at a lower level of abstraction, as for example in Déjà Vu and PARIS, is not possible either. Intractable problems should be solved globally ([Voβ '96]). In EADOCS, the complete set of design requirements is considered directly or indirectly for abstract design reasoning and reuse of complete case solutions.

In EADOCS, every case object is indexed separately for its class of structure, function or behaviour, and every case is indexed by all its objects (see section 4). A case can be retrieved for any combination of its subproblems. Design cases are obtained from previous design sessions in EADOCS, other design systems or literature. The number of cases available from literature and the number of previous design session, however, is assumed to be relatively small compared to the design space. This poses two major problems for retrieval and reuse:

© B.D. Netten, R.A. Vingerhoeds, 1997

German Workshop for Case-Based Reasoning GWCBR-97.

- The coverage of case solutions over the design space is poor, which implies that case solutions are not directly retrievable for many configurations.
- Cases have been designed for their intended application; i.e. they have been evaluated and/or optimised for a specific set of requirements in the past. These cases, however, can also satisfy design requirements that have not been defined in the case. A case solution can therefore not be related directly to its unintended functionality and behaviour.

It is highly unlikely that cases have been designed for the same set of specified design requirements. Case similarity can only be measured for the subset of the intended design problems. Computational costs for evaluation of case solutions for every new set of design requirements are unacceptably high. Retrieval of case solutions for their unintended functionality is necessary to extend the applicability of the small case-base.

The feasibility of case solutions can also be retrieved from other cases with a similar structure (see section 4). In many applications, the assumption holds that similar solutions solve similar problems. Retrieval is frequently based on the assumption in the opposite direction that similar problems can be solved by similar solutions. In EADOCS, however, similar problems can also be solved by different configurations. The initial specification of design requirements should be elaborated for the retrieval of solutions for optimal configurations.

The feasibility and optimality of design cases should be assessed for the generation of new concepts. Feasibility of new solutions can be assessed partially from their similarity with the functionality and behaviour of cases. Optimality, however, is only meaningful as a prediction for the optimised concept that can be obtained after revision of a case solution. Accurate prediction of required adaptations and their effects on optimality are not available upon retrieval of a case. Optimality of configurations cannot be retrieved from the small number of cases. A pre-selection on the configuration, based on generalised domain knowledge, is necessary to guide retrieval of conceptual designs for (near-)optimal configurations (see section 3). Preselection of configurations requires the elaboration of initial design requirements with preferences for design configurations.

3 EADOCS Design Process

3.1 Generalised domain knowledge

EADOCS is a system to support the conceptual design of fibre reinforced composite panels. The structural model is predefined; i.e. the types of panels, components, materials and geometry are defined in classes, attributes and class relations. The panel types are a set of predefined structural layouts; sandwich panels, I-, L-, T-, and Hat-stiffened panels, and corrugated panels. Panel aggregations are defined in Figure 1 using the notations from [Rumbaugh et al. '91]. A sandwich panel consists of an outer- and inner-skin and a core in between. The other panel types have only one skin and a particular type of stiffeners. The skin and flange components can be made from one layer of aluminium alloy sheet material or from laminates of one or more layers of composite materials. The fibre reinforced composite plastic (frp) materials are classified in carbon (cfrp), aramid (afrp) and glass (gfrp) fibre reinforced plastics, and are also classified in uni-directional and cross-ply materials. The laminate structure is defined in the so-called laminate lay-up, which defines the stacking sequence of layers with their material, number of plies and orientation. The type of panel, the class of materials, and the laminate lay-up are abstractions of structural solutions or prototype solutions. A concept is a design defined at the detailed level as the instance of a prototype solution.

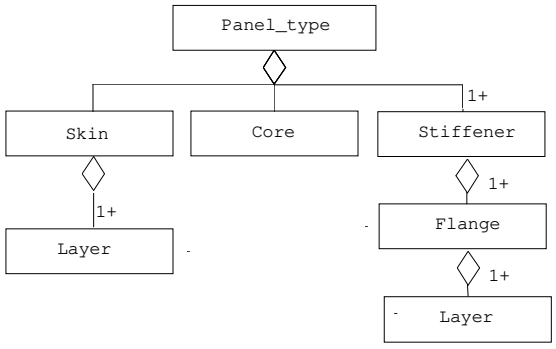

Figure 1: Structural decomposition

© B.D. Netten, R.A. Vingerhoeds, 1997

A functional required for panels is to carry loads. Usually, a design problem is initially specified by several requirements for loading conditions under different operational conditions. Each loading condition should be met by the final design. Other functional requirements can be specified for impact and fatigue resistance, or panel stiffness. Functional requirements either apply to the complete design, e.g. the loading conditions, or for particular components.

The behaviour and functionality of concepts are evaluated in accurate numerical models. Each class of structural solutions is characterised by a typical set of behavioural phenomena, for which they should be analysed. The analysis models are typical operations for classes of structural solutions. A type of panel, for example, is characterised by different modes of buckling behaviour. The models for analysing panel or component behaviour can only be applied for a completely defined conceptual design.

Abstract solutions are characterised by "prototypical" behaviour, or "prototypical" performance, for categories of design problems and optimality criteria. Prototypical performance is represented by qualitative constraints on categories of design problems. Categorisation and qualitative constraints are defined from domain knowledge. Classes of solutions are propagated through the set of constraints for a new design problem and combined as alternative prototype designs. The intended design problems of cases are classified by the categories of design problems as well. Each prototype solution is considered separately for case-based reasoning to select one or more alternative concepts that solve the specified categories of design problems.

3.2 Conceptual design process

Current design practice is to manually generate an initial conceptual design; i.e. configuration and all parameter values. Preliminary design tools exist for numerical evaluation and optimisation of each panel type. The SAPANO tool ([Van Bladel '95]) is integrated in EADOCS for the analysis and optimisation of sandwich panels. Several hundreds of design evaluations are spent for the optimisation of the initial conceptual design, while much time is spent by the designer to interpret intermediate optimisation results and to adjust the intermediate concepts and optimisation parameters. For the evaluation of multiple configurations, this process should be repeated for many different conceptual designs and several tools. This process has a large iterative loop in which many concepts are completely designed and optimised before optimality of these concepts can be assessed.

Design efficiency can be improved by an incremental approach of smaller iterations. [Pugh '81] suggested a incremental approach to conceptual design, in which alternative configurations are (manually) generated and qualitatively evaluated. The level of abstraction at which solutions are generated and evaluated should be similar, while this level is gradually decreased with each iteration. This approache is further extended in EADOCS into a progressive reasoning process of three subsequent design tasks at increasing level of detail[*] . Each task performs a more accurate search in the design space of one or more alternative solutions from the previous task. Each task is performed for one solution at a time. In the third task, only a few concepts are analysed and optimised numerically. The three subsequent tasks are:

1. **Prototype selection**

   Specifications are analysed, and abstract solutions are selected and assembled into prototypes. Prototype selection is a qualitative design process at the highest level of abstraction. Design variables and relations are represented qualitatively. All types of design variables are discretised with a similar granularity, and design functionality and behaviour are evaluated at a similar level of abstraction. The best feasible prototypes are assembled from solution classes by constraint-based reasoning about prototypical behaviour.

[*] A different implementation of three similar tasks is presented by [Bascaran et al. '89].

© B.D. Netten, R.A. Vingerhoeds, 1997

German Workshop for Case-Based Reasoning GWCBR-97.

2. **Concept selection**

Concept selection is the quantification of a prototype. Complete solutions, components and parameters are retrieved from a case-base and reused for the generation of an initial concept (see sections 4 and 5).

3. **Concept modification**

Concept modification is a quantitative process for parametric design at concrete levels. Concept configuration is not affected. Conceptual designs are modified to repair feasibility and improve performance for optimality criteria. This process is divided at two levels of accuracy. Firstly, specialist operations are applied for heuristic repair and improvement at the grainsize of discrete and integer parameters. Secondly, numerical routines are applied for more accurate and local optimisation of discrete and continuous parameters.

The task division provides several advantages for knowledge acquisition and design support. Design knowledge is available from many different sources and in many different forms. It is only possible to acquire partial models of the domain model, which can be represented and applied for specific design tasks. Design efficiency is improved by applying these partial models whenever applicable for suggesting, motivating, executing and criticising solutions for the user.

The three tasks coincide with levels of design abstraction. Abstractions of design problems and solutions are considered for configuration and generation of designs in the prototype and concept selection phases, while concrete problems and designs are considered for adaptation in the concept selection and modification phases. Adaptation of concepts is performed in three consecutive steps with increasing computational cost and accuracy; reuse of case components, heuristic modification and numerical optimisation (see also [Netten and Vingerhoeds '96]). Concept selection is performed in two steps: the retrieval of a complete case solution as the initial concept, and the retrieval of components to adapt the structure of the initial concept. Heuristics are specialist operations for the modification of a single design parameter. The numerical routines are only applicable for the optimisation of design parameters and cannot optimise the structure. Specialist operations for structural adaptation are not implemented in

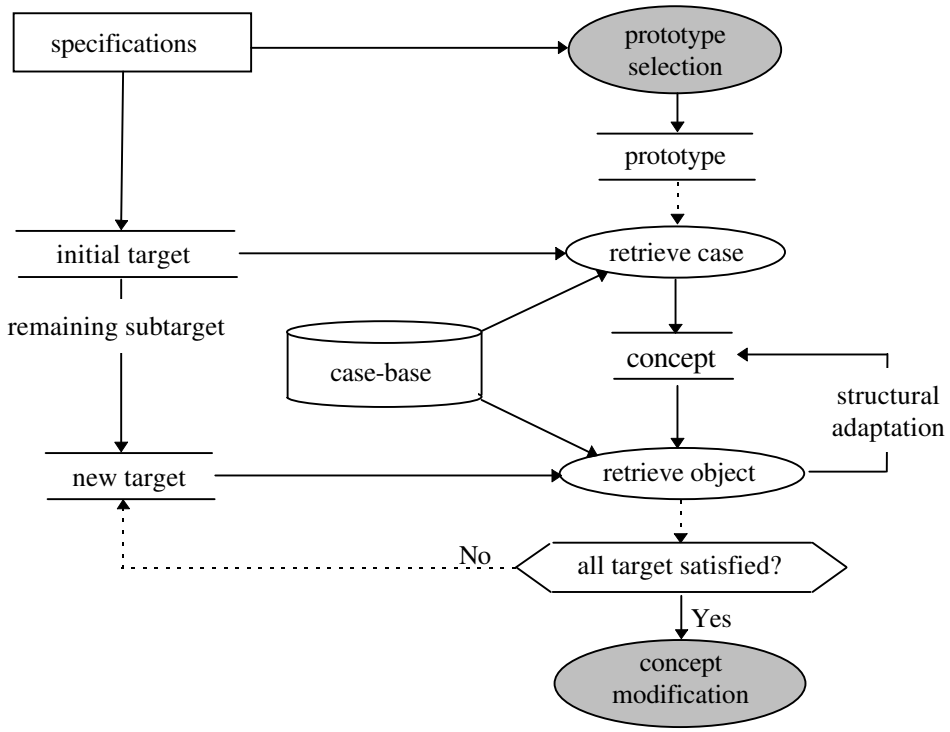

Figure 2: Concept Selection

© B.D. Netten, R.A. Vingerhoeds, 1997

EADOCS either. Optimal solutions for the design structure, and some non-numerical and discrete parameter values, should be determined during concept selection. The optimality of a selected prototype can only be assessed after conceptual design for other prototypes. Until then, the configuration of a selected prototype is assumed to identify a (locally) optimal configuration, even when a design case has non-optimal parameter values. A complete solution is required for concept modification. It is therefore important for concept selection to obtain a complete concept for the selected prototype configuration.

The appropriate abstraction level for design reasoning in each of the tasks is determined from the abstraction levels in the structural model. The behaviour and functionality are defined at the level of abstraction for the solutions. Prototypical performance is defined for each prototype solution class to distinguish its behaviour and functionality from other solutions. The differences in prototypical behaviour determine the discretisation and categorisation of design problems. Solutions for concept selection are represented more accurately by the categorisation of solutions within each solution class. Categories of solutions are defined from discretisation of parameter values. The granularity for discretisation is determined by the accuracy of the modifications and optimisations that can be made after concept selection. The categories of prototypical behaviour are refined accordingly. The categorisation of solutions and design problems is refined to a level for which concept modification can be applied to adapt the feasibility of a category of solutions within a category of design problems. In the following, only the concept selection will be discussed for structural adaptation.

4 Retrieval of an initial concept

The first task in concept selection is the retrieval of an initial concept for the elaborated specifications (Figure 2). An initial concept is retrieved as the complete solution for the case that best matches these specifications. The assumption being made here is that similar problems can be solved by similar solutions of the same prototype. These elaborated specifications are formulated as a new design problem, or initial target, for which a case solution should be retrieved. The initial target is defined by:

- A subtarget is defined for each specified design requirement. Different loading conditions for example are defined as separate subtargets. Each subtarget has a priority factor which identifies the relevance of a design requirement as interpreted by the designer for the feasibility and optimality of a design.
- Optimality of the configuration is assumed if a concept is retrieved for the prototype solution classes. Cases with other solution classes can be ignored, unless a designer relaxes the prototype.
- A prototype solution can be refined by preferences for parts of the solution. For example, the class or attribute values for some components can be defined as subtargets with a priority for their interpreted relevance.

The level of abstraction for design reasoning is determined by the categories for design problems and solutions. The target and cases are classified in these categories, and the case-base should be searched for similar cases at this level of abstraction. A classification network is built as a memory structure for retrieval in an off-line process. The memory structure represents the cases on the abstraction level of the categories. The objects of the design cases are categorised individually for their data class (structural, functional or behavioural) and by discretisation of their attribute values. Case objects are indexed and categorised individually in the memory structure. A category contains one or more case objects. The memory structure is extended for the categorisation of each design case by combination of all its categorised objects. This implies that a case can be retrieved for any combination of its design requirements, components or materials. The memory structure defines no relations between design problems and solutions.

The selection of prototype solution classes filters parts of the memory structure that contain only prototypical concepts. The remaining structure is searched for each of the subtargets individually as well as for their combination. Each subtarget is classified in the categories of problems and solutions of the memory structure.

© B.D. Netten, R.A. Vingerhoeds, 1997

The categorisation for concept selection defines the similarity measures for functionality, behaviour and solutions (see section 3.2). The operations in the concept modification phase can be applied to repair feasibility and to improve optimality of any conceptual solution in a category. The similarity of a subtarget is measured locally by its membership for each category. The similarity is scaled between 0 and 1 for a partial match and is equal to 1 for a full match. This similarity measure identifies which cases and which case objects match best for a specific design requirement. The aggregate similarity for the cases is measured by the sum of the maximum similarities of its objects for each subtarget. This sum is weighted by the relative importance factors specified for each subtarget. Aggregate similarities identify the best matching cases for retrieval of one or more initial conceptual designs for a prototype.

5 Case Combination

The retrieved case solution is a precedent for adaptation [Voβ '96]. The object similarities identify which subtargets cannot be satisfied (similarity > 0), or which subtargets have not been defined or matched by the retrieved case (similarity = 0). The absence of a similarity measure does not imply that the case solution cannot satisfy the subtarget (see section 2). A reduced similarity indicates the necessity for adaptation, such as the adaptation of the structure, or of the non-numerical or discrete design parameter values. The remaining subtargets cannot be causally related to parts of the precedent solution without detailed numerical analysis results. However, similar case solutions can identify that the current solution can satisfy the remaining subtarget, or can identify the necessary adaptations to satisfy them.

From the assumption that similar structural solutions have similar behaviour, and from the definition of categories, the following heuristics can be defined for the retrieval of structural and parametric adaptations from the case-base:

1. *Remaining targets are assumed to be satisfied when they are matched by a new case with a structure similar to the precedent solution.* Cases with a similar categorised structural solution satisfy similar categories of design problems.
2. *Differences within the categories of structural solutions can be neglected for concept selection.* Cases with a similar categorised solution require minor modifications that can be repaired or improved during concept modification (see section 3.2).
3. *The maximum set of structural adaptations can be determined from the partial matching of a new case that satisfies the remaining design requirements.*The differences in the design problem categories of the precedent case and the new case are caused by a combination of the differences in the structural solutions. The exact causal relations are unknown. The complete set of differences in solution categories is sufficient to at least satisfy the remaining design problems of the new case.
4. *Conservative adaptations can be identified to maintain the feasibility of the precedent solution.* Adaptations should also maintain the feasibility for the subtargets already satisfied by the precedent solution. The maximum set of adaptations from (3) could also affect design requirements that are already satisfied. Conflicts can be avoided by accepting only conservative adaptations from (3) that do not violate any of the specified design requirements.

The remaining target for retrieval of structural adaptations is defined by:

- All subtarget for the remaining specified design requirements.
- The prototype solution classes.
- Subtargets for every component of the precedent solution.

The case-base is searched again for the new target (Figure 2). Any case that completely matches the abstract precedent solution can identify the satisfaction of any remaining subtargets. Heuristic (1.) suggests to discard these remaining subtargets and proceed the incremental retrieval for the other remaining design requirements. The best matching case should satisfy one or more remaining subtargets for the design requirements and most of the subtargets for the precedent solution. Reduced component similarities identify the maximum required set of adaptations. These adaptations cannot be identified from the similarity measures. The relevant components should be retrieved from the case-base.

© B.D. Netten, R.A. Vingerhoeds, 1997

Conservative adaptations are only considered between two categories of the same solution class and within the selected prototype. Additional restrictions on the acceptance of adaptations can be defined from the application domain. Many design requirements are inequality constraints that can be maintained by similar constraints for conservative adaptations. An example in EADOCS is presented in the following section.

This incremental process can be continued until all design requirements are matched, or until no further adaptations can be retrieved. The validity of all heuristics for case combination should be evaluated by detailed analysis in the concept modification phase.

6 Example Design Session

Structural adaptation can be examplified in the following part of an EADOCS design session. The concept selection phase is entered for a selected sandwich panel prototype. A sandwich panel structure has three components; an upper and lower skin and a core. A skin is a composite laminate structure, which is lay-up consisting of layer components. Each layer has a unique number representing the relative position in the lay-up. A layer also has a thickness, composite material and orientation in the laminate. The laminate structure of the upper and lower skins are symmetrical. The skins are made out of carbon (cfrp) or aramid (afrp) fibre reinforced composite materials. The core is made out of aluminium honeycomb material (Al-hc).

Design requirements are specified for panel dimensions and three different panel loading conditions, each of which should be met (Table 1). Panel loads and dimensions are defined as continuous parameters, which are indexed onto categories of loading conditions and geometry. The three loading conditions in Table 1 are categorised respectively as a high compression in longitudinal panel direction, a high compression in the transverse panel direction, and a high shear load. Each loading condition and the dimensions are defined as a subtarget for retrieval. The case-base contains 20 symmetrical composite sandwich panels, of which only 7 match one or more of the subtargets. The four best matching cases have similar dimensions and satisfy only two loading conditions. Obviously, none of the available cases can be applied without adaptation.

Following background information is essential for evaluating the EADOCS approach to case combination. Two cases, identified as panels 17 and 18 (Table 3) can be combined for reuse. Panel 17 satisfies loading condition 1 and 2, while panel 18 satisfies loading conditions 2 and 3. The panels also have a similar skin laminate structure; panel 18 has an additional layer number 4 to carry the shear load of condition 3. EADOCS cannot compile this information beforehand, and retrieves this information in the following process.

The user selects panel 17 as an initial conceptual design. Loading condition 3 is not satisfied and remains as a subtarget. Loading conditions 1 and 2 are assumed to be satisfied by the retrieved panel structure and are discarded as subtargets. The structure or panel 17 is defined as new subtargets. The similarity for each target is measured by the number of similar attributes (Table 2). The aggregate similarity for the new target is the highest for panel 18, as this panel best resembles the structure of panel 17 and also satisfies loading condition 3. Component similarities identify which components of

Table 1: Initial design requirements

panel dimensions		length l = 1400 mm width w = 400 mm		
loading condition		**1**	**2**	**3**
N_x	[N/mm]	-4500	0	0
N_y	[N/mm]	0	-3500	0
N_{xy}	[N/mm]	0	0	-3000

Table 2: Local similarity of panel 18 for the new target

subtarget	**best matching layer of panel 18**	
layer of panel 17	layer number	local similarity
1	1	100 %
2	2	75 %
3	3	75 %
4	5	50 %

© B.D. Netten, R.A. Vingerhoeds, 1997

panel 18 match those of panel 17. All differences are considered for adaptation of the structure and the parameters of panel 17. Components of panel 18 are reused by substitution or instantiation using following conservative adaptations:

- The number of plies is only increased, to avoid loss of strength and stiffness of the panel.
- Layer orientations have a significant effect on panel behaviour and are not adapted. Instead, a new layer can be added to the laminate lay-up.
- Materials from the same class are never substituted. A new layer can be added using a different material.

The most significant difference between the two panels is the addition of layer number 4 in panel 18. Reuse of layer 4 from panel 18 results in the adapted concept of Table 3. The addition of the layer is a refinement of the structure of panel 17. If panel 18 would have been selected for initial retrieval in stead of panel 17, the similarity between the two panels would have indicated that the structure for panel 18 also satisfies loading condition 1.

To evaluate the improvement of the adaptation, the number of plies of both panel 17 and the adapted panel are numerically optimised for a minimum mass (Table 3). The optimised panel resulting from panel 17 is restricted to only 4 layers, because the structure of the skin laminate cannot be adapted during numerical optimisation. The optimum panel is obtained after adaptation and has a minimum mass of 3.42 [gr/cm^2].

This example shows that the approach of sections 4 and 5 can be effective for structural adaptation in EADOCS. This approach is relatively simple compared to an approach using domain dependent specialist operations. Effectiveness depends on the coverage of the case-base. For many design problems in EADOCS, no structural adaptations can be retrieved from the small case-base. Interactively, a designer can also retrieve and reuse non-prototype components, suggest structural adaptations manually, continue with alternative prototypes, or accept the initially retrieved structure and continue with concept modification.

7 Conclusions

EADOCS is a conceptual design system in which the number of available design cases is small. Design problems cannot be indexed for optimal structural solutions. Design configuration is crucial for optimality of conceptual designs. From a small case-base, the optimality of configurations cannot be assessed and domain knowledge is required to guide the retrieval to optimal prototype solutions.

Table 3 Conceptual Designs

panel structure				case-base panels		panel 17 optimised	adapted concept	adapted concept optimised
				17	18			
specific mass [gr/cm^2]				2.72	2.71	3.69	3.68	3.42
skin								
	layer nr.	ply orientation	material class	relative layer thickness [number of standard plies]				
	1	+45°	afrp	10	10	34	10	10
	2	0°	cfrp	22	9	20	22	22
	3	90°	cfrp	11	9	8	11	9
	4	+45°	cfrp	-	15	-	15	13
core			Al-hc	7	5	2.5	7	6

© B.D. Netten, R.A. Vingerhoeds, 1997

Reuse of subsolutions is necessary to improve the configuration of initial concepts, in the absence of specialist operations for structural adaptation. An domain independent approach has been presented for the incremental retrieval and composition of case parts. A number of assumptions have been identified for which the target can be reformulated for retrieval of structural adaptations. The approach is not guaranteed to work and is sensitive to the coverage of the case-base.

8 References

Bergmann R. and Wilke W., PARIS: Flexible Plan Adaptation by Abstraction and Refinement, ECAI96, Workshop Adaptation in Case-Based Reasoning, 1996.

Brown D.C., Chandrasekaran B., 1989, Design Problem Solving; Knowledge Structures and Control Strategies, Pitman, London.

Börner K., E. Pippig, E.-Ch. Tammer, C.-H. Coulon, Structural Similarity and Adaptation, In: *Advanced in Case-Based Reasonin:, EWCBR-96*, I. Smith, B. Faltings (eds.), 58-75, Springer Verlag, 1996.

Hinrichs T.R., Kolodner J.L., 1991, The Roles of Adaptation in Case-Based Design, *Proc. 9th Nat. Conf. on Artificial Intellingence*, Vol. **1**, July 14-19, 28-33, AAAI Press, Menlo Park.

Maher M.L., Bala Balachandran M., Dong Mei Zhang, 1995, Case-Based Reasoning in Design, Lawrence Erlbaum Ass. New Jersey.

Netten B.D., Knowledge Based Conceptual Design: An Application To Fibre Reinforced Composite Panels. PhD thesis, Faculty of Technical Mathematics and Informatics, Delft University of Technology, (submitted, 1997).

Netten B.D. and Vingerhoeds R.A., Incremental Adaptation for Conceptual Design in EADOCS, ECAI-96 Workshop on Adaptation in Case-Based Reasoning, 1996.

Pugh S., 1981, Concept Selection - A Method That Works. *Int. Conference on Engineering Design ICED-81*, (ed.) V. Hubka, Rome, 497-507.

Rumbaugh J., Blaha M., Premerlani W., Eddy F., Lorensen W., 1991, Object-Oriented Modeling and Design, Prentice-Hall, New Jersey.

Smyth B. and Keane M.T., Experiments on Adaptation-Guided Retrieval, *Case-Based Reasoning, Research and Development*, M. Veloso and A. Aamodt (eds.), Lecture Notes in Artificial Intelligence 1010, Springer Verlag, Berlin, 313-324, 1995.

Sycara K., Navin Chandra D., Guttal R., Koning J., Narasimhan S., CADET: A Case-Based Synthesis Tool for Engineering Design, *Int. J. Of Expert Systems*, Vol. **4**, No. 2, 157-188, 1992.

Van Bladel, P.G. (1995) Design of Fibre Reinforced Composite Panels for Aerospace Applications, PhD thesis, Fac. of Aerospace Engineering, Delft University of Technology.

Voβ A., How to solve complex problems with cases, *Engineering Applications of Artificial Intelligence,* Vol. **9** No. 4, 377-384, 1996.

© B.D. Netten, R.A. Vingerhoeds, 1997