523-61
136891
P8

Punch
N93-17522

A Domain-Specific Design Architecture for Composite Material Design and Aircraft part Redesign

W. F. Punch III[1], K.J. Keller, W. Bond and J. Sticklen[2]

1 Polymer Composite Materials

Advanced composites have been targeted as a "leapfrog" technology that would provide a unique global competitive position for U.S. industry. Composites are unique in the requirements for an integrated approach to designing, manufacturing, and marketing of products developed utilizing the new materials of construction. Numerous studies extending across the entire economic spectrum of the United States from aerospace to military to durable goods have identified composites as a "key" technology.

A typical chronology for designing a composite material is as follows [5]. First, macroscopic properties which are desired in the completed composite are set. Properties such as final material tensile modulus, resistance to acids and alkalis, and electrical resistance are parameterized. Based on these desired properties, the composite designer proposes an initial plan for the production of the composite. This plan includes both an ingredients list for all materials to be initially present, and a preliminary protocol which states how the initial mixture is to be processed. Next, the composite designer estimates how well the proposed composite design meets the initially stated, desired properties. This estimate is carried out along two paths. The composite designer may actually produce samples of the composite, then perform laboratory testing to determine properties of interest. Or the designer may model the proposed composite to estimate its properties. Ultimately, a proposed composite design will result in an actual material which can be subjected to laboratory testing. But, one goal of composite researchers is to provide better models for proposed designs in order to limit the number of candidate materials which must actually be fabricated for testing. Following one round of design proposing, and matching to specifications, successive rounds of redesign are usually required before convergence of proposed composite properties to desired properties takes place.

In general there have been two approaches to composite construction: build models of a given composite material, then determine characteristics of the material via numerical simulation and empirical testing a(leaders along this path include the research groups of Dr. Larry Drzal and Dr. Martin Hawley, Michigan State University Composites Center), nd experience-directed construction of fabrication plans for building composites with given properties (e.g., the recent work of Frances Abrams at AFWAL Materials Lab at Wright Patterson). The first route set a goal to capture basic understanding of a device (the composite) by use of a rigorous mathematical model; the second attempts to capture the expertise about the process of fabricating a composite (to date) at a surface level typically expressed in a rule based system. The first important point is that, from an AI perspective, these two research lines are attacking distinctly different problems. Secondly, both tracks have current limitations. The mathematical modeling approach has yielded a wealth of data but a large number of simplifying assumptions are needed to make numerical simulation tractable. Likewise, although surface level expertise about how to build a particular composite may yield important results, recent trends in the KBS area are towards augmenting surface level problem solving with deeper level knowledge.

1.1 Redesign of Existing Metal Parts from Composites

Utilizing composite parts in engineered devices offers multiple advantages over the use of traditional metal parts. These include weight savings, strength::weight ratio increase, and greater flexibility in processing. Because of these relative advantages, there is great interest in retrofitting existing metal components with composite material counterparts. However, although such retrofitting is under way, many of the advantages of composite materials are not being fully exploited. Current practice throughout many industries is to take an existing metal part, and try to redesign it piece wise as a set of very simple composite material

---

[1] Author to whom correspondence should be sent.

[2] Punch and Sticklen are with the AI/KBS Lab., Computer Science Department, Michigan State University, East Lansing, MI 48824. They can be contacted at punch@cps.msu.edu and sticklen@cps.msu.edu, respectively. Keller is with the AI Center, Bond is with the McDonnell Douglas Research lab, of the McDonnell Aircraft Company St. Louis, MO 63166-2995

components. From a design perspective, given present state of the art, this is a reasonable practice: the larger the composite part, and the more geometrically complex, the more difficult the design task becomes.

Many of the relative advantages of composites; e.g., the strength::weight ratio; is most prominent when the entire component is designed as a unitary piece. The bottleneck in undertaking such unitary design lies in the difficulty of the re-design task. Designing the fabrication protocols for a complex-shaped, thick section composite are currently very difficult. It is in fact, this difficulty that our research will address.

2 The Design Problem

In redesigning an existing aircraft part with a new composite part, the composite engineer is faced with numerous design and manufacturing options with interrelated effects on all aspects of the product lifecycle (strength, weight, producibility, survivability, maintainability, and supportability are some of the issues). Capturing and managing these interrelationships is necessary for successful implementation of an optimum design environment.

This process requires the integration of knowledge and data from many disciplines residing on multiple platforms and repositories. Most design decisions have interrelated effects which are extremely difficult to predict. For instance, a limited knowledge of composite materials could result in a design that will not meet temperature requirements, is improperly layed-up, will not conform to the manufacturing process, or results in a high scrappage rate when manufactured. Redesign of an existing part is more complex since it also requires the designer to reconstitute an understanding of the functions of the original part.

An intelligent design system which could integrate and arbitrate knowledge and calculations obtained from multiple sources would be a valuable resource to both the inexperienced design engineer who must design "simple" parts and also to the experienced engineer who is focusing on more complex designs. This system must be flexible and expandable to accommodate advances in composite material design and manufacturing methods and to facilitate the incorporation of knowledge and calculations from multiple sources. We plan to meet these challenges by developing a framework which can bring appropriate problem solving techniques to bear at the correct time.

2.1 Testbed Design Complexity

Our initial works concentrates on "simple" composite parts. The processes required to produce these parts are generally well known and proven. In addition, the parts that fall into this category are numerous. Simple parts include clips, brackets, avionic shelves, doors, etc. Complex parts such as outer moldline skins and internal bulkheads, frames, and longerons typically account for only 10 percent of the number of parts that go into an airframe.

The current design process is outlined in Figure 1 which describes the inputs, the design process, and the desired outputs.

To automate these tasks, the overall system architecture shown in Figure 2 is anticipated.

2.2 Testbed Design Architecture

The system will provide support to the design engineer in a number of different ways:

- Interactive design advice: will check input design parameters and part data for errors and design rule violations. If the feature violates a design rule, the system informs the designer.
- Process and material selection: provide the user with predictions of the cost and probability to successfully produce a given part and/or engineering feature. These predictions can be based on knowledge-based results or on analytical process simulation models.
- Analysis support: provides a method to obtain strength information based on closed form solutions. This would permit a designer to perform quick strength predictions before carrying out complete finite element analysis.

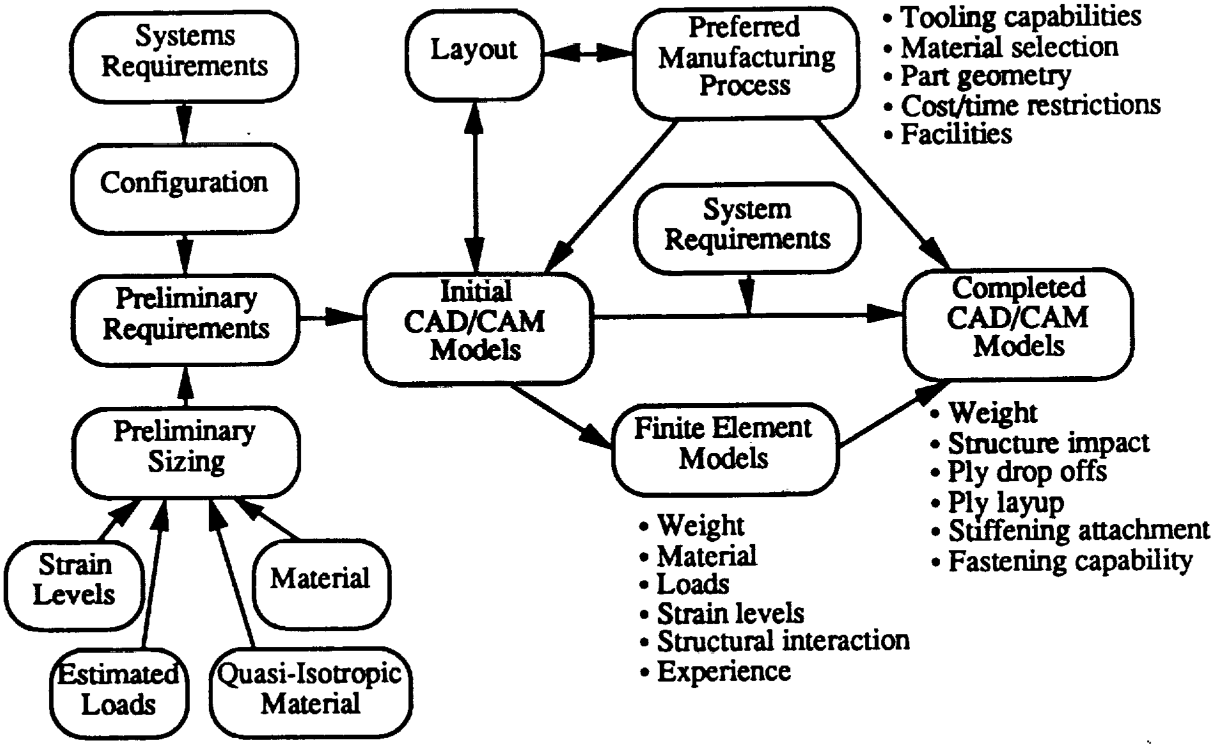

Figure 1: The composite design process

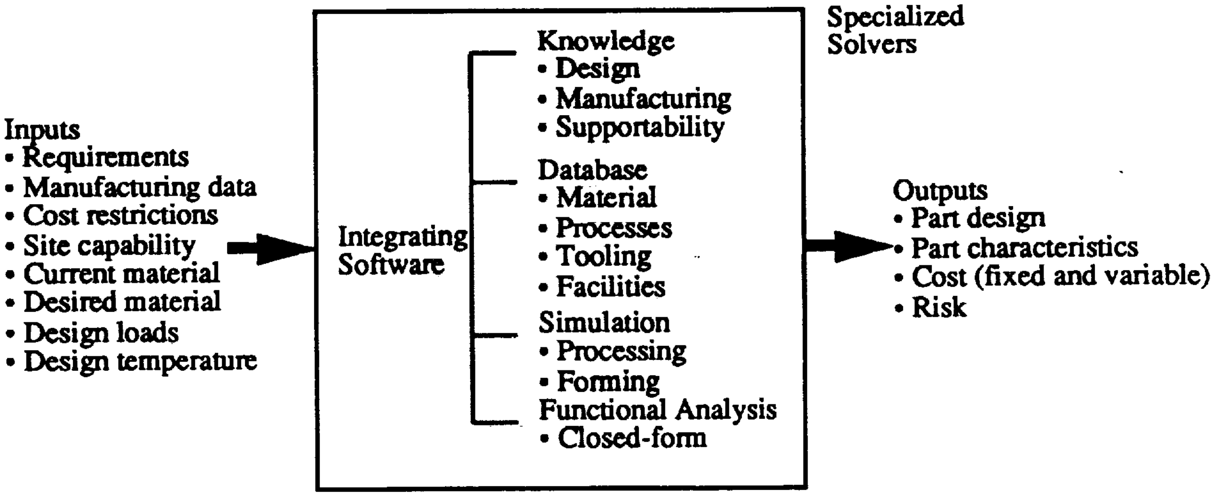

Figure 2: The expected system architecture

- Process plan support: produce process documentation by completing standard templates based on design and manufacturing rules (a good example is a ply orientation table based on standard design rules for given material and strength requirements)

3 Role of AI in the Composite Materials Redesign Domain

In the composite materials redesign area, the heterogeneous problem solving techniques which we elaborated above are all required to produce robust problem solving results. In addition, we are investigating the integration of other, more well explored problem solving methods which include:

- *design problem solving:* - Design problem solving will play an important role in our application domain. We intend to utilize Routine Design (Brown & Chandrasekaran, 1986) as a starting point - similar to the applications of Sticklen et. al. (Sticklen,Kamel,Hawley, & Delong, 1991). In addition, we may utilize a case base of known designs which must be altered in minor ways to meet requirements - much in the spirit of the work of Goel on redesign using a functionally indexed case base (Goel & Chandrasekaran, 1989).
- *capturing design rationale:* - As pointed out above, current practice for the problem of redesign of metal parts from composite materials is hampered by concentrating on small scale replace, rather than entire component redesign. A major reason for this is that the purposes of the component to be redesigned are not available. We will meet this problem by first capturing the component purpose utilizing a *Functional Reasoning* (FR) Approach - following the work of Bond and Sticklen in the aerospace domain (Bond,Sticklen, & Pegah, 1991; Pegah,Bond, & Sticklen, 1991; Sticklen,Bond, & St. Clair, 1988).
- *simulation in the service of design verification:*- Once a composite materials-based redesign for a component is proposed, there is a need to "test" it. This will be undertaken using a combination of FR+bond graph simulation. A functional representation augmented with primitives in bond graph will form the basis for a simulation of the *process* of composite material curing. This simulation will yield the characteristics of the cured composite produced by following the proposed design.

4 Integration Architecture

A pivotal part of our research lies in the integration architecture which will allow the smooth interaction of both disparate problem solving agents, and will allow the access and use of heterogeneous knowledge and data bases. For our initial exportation, we will apply the Task Integrated Problem Solver (TIPS) approach [3, 4, 2]. Characteristics of TIPS which we are important for domain-specific design are:

1. TIPS provides dynamic integration. That is, the which problem-solver is invoked is determined by the problem state, previous problem-solving history, knowledge available and other factors. The "chunk" size of the methods that TIPS supports is higher than than of SOAR and this is helpful for applications where much knowledge is compiled with nevertheless need for runtime integration.
2. TIPS exploits the task structure, ie. the goal-subgoal structure of the overall problem, to identify methods that might be relevant.
3. TIPS can mix different types of problems solvers. The problem-solvers need some minimum of communication capabilities, however, but this can be added to the kernel problem solver.
4. TIPS is capable of supporting task-level explanation of why a particular problem-solver was invoked.

The basis for the representation of control used in TIPS is the Sponsor-Selector system first used in DSPL (Design Specialists and Plans Language) [1]. It consists of a hierarchy of three parts: at the top a *selector*, under the selector some number of *sponsors*, and under each sponsor a *method invocation*. In short, the available problem-solving methods are grouped under the selector as sponsor-method pairs, where each sponsor provides appropriateness measures for its associated method invocation. At any control choice point (i.e., some point in the flow of problem-solving at which another method could be invoked) the overall control process is to run all the sponsors to rate their associated methods, then have the selector choose the next method to be executed based on the sponsor values and other data.

The sponsors are therefore used as "local" measures of how appropriate a problem-solver is for achieving the current goal, while the selector takes a more "global" view of selecting which of the available methods is the "best" under the present circumstances. Both the sponsors and selectors encode their knowledge in a patter-match table that indicates what features are important for making the decision and how those combinations of features contribute to the final answer (selection or appropriateness measure).

The TIPS architecture has been used to implement a large-grained medical diagnosis system in the domain of liver and blood disorders [4] integrating Compiled/Association-Based Diagnosis, Causal Reasoning, Data Gathering, Data Validation, Therapy Planning and User Interaction.

4.1 Integrated Reasoning in the Composite Domain

Consider the problem as presented by a portion of Figure 1. `Configuration` and `Preliminary Sizing` are two methods whose results can be used by the method `Preliminary Requirements` to set up "rough design" parameters in the early stage of the design. In fact, the representation as listed could be more complicated, `Preliminary Requirements` might require an number of invocations of both the `Configuration` and `Preliminary Sizing` methods to reach a stable configuration where each invocation requests small modifications to the initial answers. Each new invocation would contain information about the new problem that `Preliminary Requirements` has perceived, perhaps even some suggestions about how to repair the problem, and asks for further refinement. Repeated invocations continues until the `Preliminary Requirements` method has achieved its goal, and problem-solving then continues to more detailed designing. The*dynamic* interaction between multiple methods as shown in this example is the kind of problem-solving that an integrated reasoner has to capture to be effective in this domain. That is, determining when a method is done depends on the state of the present problem and the perception by the system of which goals are "active" and if they have been achieved.

5 Future Directions, Extending the Integration Architecture

The first problem is one of representation. Figure 1 is a direct representation of the methods used to achieve goals in the composite design problem, but not the goals implicit in guiding the problem-solving. In the current TIPS implementation, the methods are directly represented, but the goals are only represented implicitly in the sponsor-selector system.

The second problem is one of standardizing the means by which sponsors can monitor goal status and by which methods can indicate their success, partial success or failure. Likewise, a common means by which problem state information is gathered must be made available. At present, Lisp code specific to the goals and methods in the diagnostic system have been used but this needs to be supported by an architectural feature.

Both are important additions to a TIPS architecture. Direct representations of the goals of a domain make the job of mapping an analysis directly to code much simpler. It also enhances other aspects of a system, such as explaining why certain steps were taken in a case run (because the goal situation at that stage was X and Y was the best choice etc.). Standardized interactions between methods is also quite important will enable cooperation and negotiation among problem-solving methods to solver larger problems. These problems and others will form the basis for research on domain-specific design in the years to come.

References

[1] D. Brown and B. Chandrasekaran. *Design Problem Solving: Knowledge Structures and Control Strategies.* Pitman, London, UK, 1989.

[2] W. F. Punch III. *A Diagnosis System Using a Task Integrated Problem Solving Architecture (TIPS), Including Causal Reasoning.* PhD thesis, The Ohio State University, 1989.

[3] W. F. Punch III. TIPS (task-integrated problem solver), a task-specific integration architecture for heterogeneous agents. In *Proceedings of the AAAI-91 Workshop on Cooperation Among Heterogeneous Intelligent Systems*, pages 1–10, Anahiem, CA, July 1991.

[4] W. F. Punch and B. Chandrasekaran. An investigation of the roles of problem-solving methods in diagnosis. In *Proceedings of Second Generation Expert System's Conference*, pages 25–36, May 1990.

[5] V. Venkatasubramanian, Y. Lee, and C. Gryte. Design of polymer composites: A knowledge-based framework. In *AIChE87*, 1987.