PROGRESS REPORT
FOR YEAR 2 OF THE RESEARCH GRANT ON
AI AND DESIGN: COMPUTER AIDED PRODUCTIVITY (CAP)
AND PLAN FOR YEAR 3

Saul Amarel, PI
Louis Steinberg, Co-PI
Tom Ellman
Andrew Gelsey
Haym Hirsh
Jerry Richter

1. Introduction

During the current period, a major part of our effort was directed towards **implementation of our system testbeds**, primarily **the Design Associate for ship design**, and also the **FAD/ASP system for microprocessor design**. Some of our **core work** was **closely related to these system testbeds**, and in addition we continued our **core work in approximate modeling of thermal systems and several other areas**. The work on Knowledge Compilation that was being done by Dr. Chris Tong continued but was supported primarily by a separate NSF grant to Dr. Tong.

We also continued developing contacts with potential collaborators in other, related, design domains, including the **design of small electromechanical parts** (with Dr. Larry Leifer and his colleagues at Stanford, and with Dr. Catherine Baudin at NASA Ames), the **design of aircraft jet engines** (with Dr. Siu Tong and his colleagues at GE), and the **conceptual design of aircraft** (with Dr. George Kidwell, Dr. Hiro Miura and their colleagues at NASA Ames).

The next section will outline the perspective from which we are working, discussing how we see our project fitting in with, and extending, the current state of research on knowledge-based design. The following two sections will talk about the system testbeds we are developing and their respective related core work. The section after that will discuss the other core work, and the final two sections will, respectively, report our dissemination, planning and infrastructure development activities for Year 2 and cover our research plans for Year 3. An appendix lists external publications from our group in the past year.

2. Extending the Frontiers of AI and Design

The ultimate goal of our work is a ‘‘Science of Design’’, i.e., a set of domain-independent principles and techniques for doing design tasks. In particular, we are interested in adapting and extending principles and techniques from AI for building computer systems which do, or help people do, design. However, early in our work it became clear that there is very little that can be said that applies to *all* design tasks. Designing a spoon, for instance, is quite different from designing a digital circuit. We believe that there are a number of dimensions along which design problems vary, and the appropriate techniques for a problem depend in systematic ways on where a problem falls along these dimensions [8, 4]. We can view these dimensions as inducing a space of design tasks. Our ultimate goal is to understand what these dimensions are and how to determine where in this space a task fits, to develop techniques and principles that apply to *regions* of this space, and to understand the mapping that tells us which techniques apply to which region.

The current efforts of the CAP project can be seen as exploring a region of the design-task space that are of great practical importance but have not been the primary focus of most previous work in the field. We are pushing the state of the art along three dimensions:

- Availability of knowledge for reasoning directly from goals/constraints of the design task to parts of a solution (a design). An important special case is knowledge that permits decomposition of a design task into nearly independent subtasks. We are concerned with cases where initial availability of this type of knowledge is low.
- Cost of evaluating a complete design. We are concerned with cases where evaluation cost is high.
- Completeness of the problem formulation, both in terms of the search space and in terms of the specification of the problem conditions, i.e., the design goals and constraints. We are concerned with cases where completeness of problem formulation is low.

One of the most significant dimensions for categorizing design tasks is the degree to which the design task can be recursively decomposed into separate subproblems, where each can be solved and the solution evaluated independently. If such decomposition is possible, we have what has been called a *derivation problem* [1], and we can work in a systematic way from the specifications of the problem to its solution. Most work on AI in design (including our own [9]) has focused on tasks that can, to a first order approximation, be treated as decomposable into mostly independent subproblems. Since this is only an approximation, the subproblems do interact somewhat, and additional mechanisms (e.g., constraint propagation) are used to handle these interactions, allowing us to search in a space of partial designs, gradually building one that is complete.

However, we have found that such mechanisms are effective only if the constraints are *local*, that is, if they involve at most a few subproblems involving a small part of the artifact. When constraints are *global*, involving large numbers of decisions and large parts of the artifact, constraint propagation and other such mechanisms break down, taking large amounts of computation and providing little pruning of the search space.

Thus, when global constraints are important, there is no general way to reason from the specifications to the solution, and we have what has been called a *formation problem* [1]. The only obvious approach is to propose a complete candidate design, evaluate it, and, if it is not acceptable, modify it and try again. In other words, since constraints are global, there is no way to evaluate a part of the design by itself, without having designed the other parts. Only if you have a complete artifact can you evaluate it, so we must search in a space of complete designs rather than partial designs.

Unfortunately, global constraints are the norm, rather than the exception, in many real-world design tasks. They arise from two sources. First of all, in some domains many parts of the artifact interact physically. A prime example of this occurs when the design involves hydrodynamics. A flow field around an object is the result of the global shape of the object and different parts of the object interact in their effects on the flow.

Secondly, and more importantly, real-world design usually involves issues of ‘‘how good’’ an artifact is, and not merely whether it functions correctly. How good an artifact is usually depends on how much it uses various global *resources*, e.g. how much time, power, and silicon area a digital circuit takes, how much drag a hydrodynamic shape causes, or in general how much money it will cost to make and use the artifact. If there is a limit, say, on how much power a circuit can consume, then each microwatt consumed by one part of the circuit is that much less power available for the other parts, and so all design decisions interact if they affect power anywhere in the circuit.

A few pieces of work in AI and design have looked at problems with global constraints, e.g [5, 6, 7], but these have looked at problems (e.g. designing a single gear pair), and where evaluating a design is a very cheap computation. This brings us to the second dimension of the design space where part of our work (the work on boat design) is pushing into new regions: the cost of evaluating a complete design. In many real-world design tasks, the cost of evaluating a candidate design is a central issue. For instance, to evaluate the hydrodynamics of even a single ship hull using all the known physics is beyond the scope even of a modern supercomputer. The advent of tera-op computing will make such computations feasible, but even so there will still be a tradeoff between the amount of time spent evaluating each candidate and how much of the design space can be searched. In such domains a practical design process must involve finding and using approximate models for most of the search, and thus must also involve methods for *choosing* the right model to use at each point.

Furthermore, in such domains the process of modeling a physical system is often more of an art than a science; it is very easy even for humans to misuse a modeling program and get results that are meaningless. Thus, in many domains, two essential issues in design are (1) how to use models of the artifact and of the physics involved in order to evaluate the artifact relative to the task goals, and (2) how to be sure the results of this evaluation are meaningful.

The third dimension we are interested in exploring is the degree to which the search space and the specification of what constitutes an acceptable design are only partially formulated when the design task starts. These issues arise especially in the the early phase of design often called "conceptual design". In this phase, the set of alternatives to consider is not completely known, and the desired result is not so much a single design as it is knowledge about what the possibilities and tradeoffs are. Another kind of task where such incomplete problem formulation occurs is in problems such as those requiring the design of a free-form shape, where the space of alternative is simply too large and unstructured to be completely searched, and part of the process of formulating the design problem involves choosing a subspace in which to search.

Because most previous work has focused on problems with primarily local constraints and/or problems where approximate evaluation and modeling are not central issues, we believe that the methods that have been developed will not be sufficient when we scale up to most real-world problems.

Therefore, we have chosen to focus our efforts on two design tasks: the hydrodynamic design of a sailboat hull and, secondarily, the design of a single-chip microprocessor from an Instruction Set Architecture (ISA). Both of these are real-world problems; we are not using toy versions of them. Both of them (in their real-world form, at least) crucially involve global constraints - sailboat design because it involves fluid flow around a shape, and both of them because they involve global properties of the design (drag in the case of sailboat hulls, and time, power, and area in the case of microprocessors). In sailboat design, in particular, evaluation and modeling are central issues, and issues of formulating and structuring a search space also arise.

3. Hydrodynamic Ship Design

A major portion of our research has focused on the design of large, complex, physical structures, using the domain of hydrodynamic design of sailing yachts as a research testbed. Participants in this area include Drs. Saul Amarel, Louis Steinberg, Thomas Ellman, Andrew Gelsey, Haym Hirsh and Jerry Richter, as well as graduate students Ke-Thia Yao, John Keane and Mark Schwabacher, and our programmer, Tim Weinrich. Our research strategy has been to **quickly develop a fairly complete system**, both to guide us as to what problems need further study, and to provide a testbed for evaluating the approaches to these problems that we are developing . **This initial system is now running.** In parallel with developing the testbed system, we also began looking at a set of specific research issues; now that the initial system is running we are turning even more of our attention to these issues.

Hydrodynamic design is an especially difficult problem because of the complexity of the interactions between parts of a shape and the overall flow field it causes. It may not be possible to do much decomposition, or even much blame assignment (e.g., determining how much a given part of the surface contributes to lift and drag), beyond the gross decomposition of a hull into the keel and the "canoe body" (the rest of the hull without the keel). Thus, there may not be a better approach to designing the hull than a fairly simple one of hill climbing using a structured set of operators. Our Design Associate takes this approach - to design a boat for a set of new conditions (race course and wind speed) it starts with a previously designed boat from a library and then iteratively modifies this boat's shape to improve its performance for the new conditions.

However, in our discussions with domain experts, it has become clear that much of their effort goes, not into the hillclimbing itself, but rather into a set of preliminary studies that provide the information needed to set up and guide the hill climbing process. That is, much of the work goes into **formulating the search problem**. This involves choosing a starting point (an initial design), choosing a set of operators for each stage of the design, and choosing an appropriate evaluation function (to determine the performance of the candidate designs) for each stage.

Thus, while we have implemented a hill climber as a necessary part of our testbed, we view our primary contribution as lying not so much in the hill climber itself as in automating the initial, offline process which formulates the search problem, and in automating the selection and use of evaluation functions during the search.

So far, most of our work on formulating the search has focused on formulating the set of operators that are used to modify the shape of a boat. We have some ideas about how these operators might be automatically discovered, but our focus has been on taking a set of manually-generated operators and automatically **discovering and using decompositions** of this set into relatively independent subsets. We also have some ideas, not as yet implemented, on how to choose the initial design and how to expand the design space to include innovative structures. In the terms used in our original proposal, this work lies in the general area of **decision making**.

We have also put substantial effort into the issues of **selecting the right approximate model** at each point in the design process, and **the use of AI to support robust simulation**, i.e., to automate the process of getting meaningful results from complex numerical simulations. This includes also the process of finding appropriate panelizations for solving numerically PDE's that enter into the hydrodynamic analysis of candidate designs. Finally, we have done some initial work on developing good numerical strategies for applying computational fluid dynamics codes to geometry optimization. In the terms of our proposal, this work lies in the general area of **modeling**.

We will first discuss the overall architecture of our system, and then discuss the specific issues we have focused on and some where we have ideas we would like to pursue.

3.1. The Design Associate

The effort to develop our initial Design Associate was led by Dr. Thomas Ellman, with substantial help from John Keane, Mark Schwabacher and Tim Weinrich. We also had numerous discussions with our collaborators Nils Salvesen and Marty Fritts of SAIC, and John Letcher of Aero-Hydro Incorporated, aimed at characterizing the principal phases of the design process as manifested in the design of the Stars and Stripes ’87 racing yacht. On the basis of these discussions, we have **defined the overall architecture of the Design Associate**. This architecture includes facilities for managing several phases of the design process:

1. Goal-directed retrieval of an initial design
2. Structuring of design space, and multi-level search of the space for a (near) optimal design, i.e., a design that best satisfies the design goals and constraints
3. Intelligent selection of suitable evaluation models
4. Impasse-directed structural innovation.

We have **implemented a complete initial version of the design associate** that realizes most elements of this architecture. (The initial design is selected manually at the moment, and the innovation feature is not implemented yet.) This version **runs, and is able to do a complete job of hydrodynamic hull design** for racing yachts, although some components of the architecture are still in quite rudimentary forms.

3.2. Structuring of design space; Discovering and Using Decompositions

The design associate is currently being used as a testbed for our core research on automatic decomposition of design problems. Main participants in this work have been Drs. Thomas Ellman and Haym Hirsh, and Mr. Mark Schwabacher.

Decomposition is especially important in the design of complex physical shapes such as yacht hulls. Exhaustive search for solution is totally unacceptable because hull shapes are specified by a large number of parameters. By partitioning the shape parameters into non-interacting or weakly-interacting sets we can achieve an effective decomposition of the search space, and thus a substantial reduction in the cost of search. We have developed a combination of empirical and knowledge-based techniques for finding useful decompositions. The knowledge-based method examines a declarative[1] description of the function that relates design structure to design goals/constraints, in order to identify parameters of the design that potentially interact with each other - relative to design goals. The empirical method runs computational experiments in order to determine which potential interactions actually do occur in practice.

Each computational experiment measures the degree to which two parameters interact. Design

[1]as opposed to only executable

parameters are combined into subsets such that the interaction between any two parameters in different subsets is less than some threshold. Then, search is carried out in a series of phases, with each phase concentrating on changes in a single subset of design parameters.

We expect this approach to finding decompositions will result in faster search for solution, with a minimal sacrifice in the quality of the resulting design. **An initial implementation of this approach has been carried out; we are currently revising this implementation and planning a set of computational experiments to be run in the coming year to test the usefulness of these ideas.**

3.3. Intelligent Model Selection for Search in Design Space

The design associate is also being used as a testbed for our core research on intelligent model selection in the search for a design that satisfies (near) optimally the given design goals/constraints. Main participants in this work have been Dr. Thomas Ellman, and John Keane.

The model selection problem results from the difficulty of using exact models to analyze the performance of candidate designs. For example, the lift-induced resistance of a yacht keel can be predicted by solving La Place's equation, or by using a simple algebraic formula. The two approximations differ widely in both the costs of computation and the precision of the results. Intelligent model selection techniques are therefore needed to determine which model is appropriate at each point in the design process.

We have developed a technique which we call **gradient magnitude based model selection**, to perform model selection in the context of hillclimbing search. This technique is based on the observation that when climbing a steep slope, even a highly approximate model will often suffice to determine which way is uphill, because candidates being compared will be very different. On a more gradual incline, however, the candidates will be much closer to each other and to the current design in their values under an evaluation function that estimates degree of goal attainment, and a more precise model will be required to tell which is better. Our technique operates by comparing the estimated error of an approximation to the magnitude of the local gradient of this goal attainment function. An approximation is considered acceptable as long as the gradient is large enough, or the error is small enough, so that each proposed hillclimbing step chosen according to the approximation is guaranteed to improve the actual value of the goal attainment function. **Implementation and testing of this approach are currently in progress, and will be continued in the coming year.**

3.4. Using AI to Support Robust Simulation

We have also looked at the use of AI to support robust simulations, i.e., to automate the process of getting meaningful results from complex numerical simulations. Main participants in this work have been Drs. Andrew Gelsey and Jerry Richter, as well as graduate student Ke-Thia Yao.

Computational simulation is an important tool for predicting the behavior of physical systems, but using present day simulation programs requires considerable human effort and expertise to

- set up the simulation by transforming a description of the physical situation into a representation the simulation program can successfully process,
- analyze the output of the simulation program to extract desired information and in particular to determine whether the output makes sense and how accurate it is likely to be, and if the output is not acceptable, to

- determine how to change the simulation program's input so that it will more reliably predict the behavior of the physical system being analyzed

Automating this process will require significant progress in two AI research areas, i.e., in spatial reasoning and deep models of expert reasoning about physics, and in numerical analysis. The simulation program we've been using is PMARC[2], a "panel method" which predicts fluid flow around a solid body (like a sailing yacht) using a partition of the surface of the body into distinct "panels" on which the program computes a value called the velocity potential. Spatial reasoning is essential to find geometrically plausible input for PMARC. For example, panels must lie on the surface of the body and cover the entire surface. However, many spatially plausible panelizations may produce wrong or unreliable fluid flow prediction. Choosing a good configuration of panels from the set of possible configurations requires significant reasoning about both physics and numerical analysis.

Our initial implementation effort has focused on extending the standard version of the PMARC simulator so that it can automatically set up PMARC computations for a limited range of physical situations. This first step is important for the following reasons:

- By doing this process ourselves, we are learning more about the kinds of physical, spatial and numerical reasoning that we will need to automate for the system to handle a more general range of physical situations.
- This implementation provides an interface allowing automated control of PMARC by another program, which will be essential for exploring our ideas about spatial reasoning and modeling physical systems
- This implementation allows the Design Associate to automatically invoke PMARC to compute a yacht's effective draft, which is essential for testing ideas being developed in that work for choosing between PMARC and cheaper but less accurate models.

Our extension of PMARC involved adding the following capabilities:

- automatic panelization using an algorithm based on solving appropriate sets of nonlinear equations
- automatic generation of a wake which follows streamlines computed by PMARC
- computation of effective draft by wake integration

We have now successfully implemented these capabilities. During the coming year, we plan to pursue our ideas regarding spatial reasoning and deep models of expert reasoning about physics and numerical analysis. In the spatial reasoning area, we are attempting to formulate the PMARC panelization problem as a search for appropriate transformations within a hierarchy of spatial representations for a yacht (or other physical object). In the modeling area, we are attempting to formalize the space of approximate physical models on which PMARC is based, in order to make all simplifying assumptions explicit so that PMARC output can be tested for consistency with these underlying assumptions.

---

[2]Panel Method Ames Research Center

3.5. Geometry Optimization Using Computational Fluid Dynamics

We are developing numerical strategies for applying computational fluid dynamics codes in a design setting where an optimal geometry is sought. Main participants in this work have been Dr. Gerry Richter and Mr. Ke-Thia Yao.

Most of our work has been done in the framework of panel methods for potential flow. Frequently, one is interested only in gross features of the flow field, e.g, forces and moments. We have developed an efficient procedure for assessing how these will vary with small changes in the geometry. The standard process of solving the flow equations involves taking a matrix of coefficients describing the panels (discrete segments of the surface of a shape) and reducing this matrix to its triangular factors (triangular matrices whose product is the original matrix). This process is the main computational cost of solving the equations. We have developed a way to take these matrices and efficiently calculate not only the quantities of interest (e.g. the forces), but also the derivatives of these quantities with respect to changes in the shape of the surface. These derivatives can then be used to guide the shape optimization process.

This approach is cheaper than the more obvious approach of determining the derivative by perturbing geometry, solving the resulting flow problems from scratch, and seeing how much the global quantity of interest changes. The cost of finding each derivative is reduced from $O(n^3)$ to $O(n^2)$ where n is the number of panels. (For the keel example, we need roughly n = 1000 panels.)

Our initial exploration shows that the basic idea works, and we would like to apply it in a more general design framework. However, we have been having some trouble getting PMARC to give sensible results even for the initial solution. We are working on understanding why this has happened, and plan to continue this investigation in the coming year, and eventually integrate this work into the Design Associate.

3.6. Retrieving Designs from a Library and Introducing Structural Innovation

Two areas we would like to pursue in depth in the coming year are (1) how to automatically retrieve a "good" initial design from a library of previous designs, and (2) how to automate the introduction of structural innovation.

As discussed above, the Design Associate starts off by retrieving a previous design from a library. This initial design is not just the starting point for the design process. Because the set shape modification operators associated with an initial design is given, and it is restricted (i.e., the operators cannot in general turn any arbitrary shape into any other), the starting point also determines the space of shapes that are reachable from the initial design, i.e. the design space. Because of this, it is particularly important to choose the right one.

Currently the selection is done manually, but we plan very shortly to implement a simple automatic method. The approach will be to simply evaluate each boat in the library against the new specifications - e.g. the conditions on cost, time around the given race course under the given wind, and "rating" (e.g. the 12-Meter Rule) - and choose the boat that comes closest to meeting the new specifications. This approach is appropriate for a number of reasons:

- We do not expect to have large libraries. E.g., the most extensive hull shape library we know of has only a few hundred entries.
- It is not expensive to evaluate a boat against new specifications. Even for time-of-traversal

> of a race course for a given wind, once a boat has been evaluated on one race course and wind speed, enough information is available (and can be stored in the library entry for the boat), that evaluating performance on another race course or wind speed is quite cheap.

However, we believe this approach may not be the best possible one. Consider the fact that the speed of a boat is determined by a balance of a number of forces, including friction, and lift forces on the sail and keel. Design modifications that improve one force may make another worse; a good boat is one with the right set of tradeoffs between the forces. Furthermore, the right tradeoff depends on the specifications to be met. E.g., for a race that is primarily downwind, keel lift is much less important and skin friction more important, than for one that is more upwind.

Thus, while the overall performance of one boat may not be as good as that of another relative to the new design specifications, it may be a desirable place from which to start the search for a design that satisfies the new specifications because it embodies a particularly good tradeoff between some set of the forces. It may be possible to modify the boat in a way that maintains this good tradeoff while adjusting the tradeoffs between other forces, and thus reach a superior boat. We would like to pursue the idea of recording and using such tradeoff information for library retrieval.

We would also like to pursue the idea of using such tradeoff information as a guide to structural innovation. The Design Associate currently has no way to revise the *structure* of the boat it is designing, that is, to add additional parts such as the addition of winglets to the keel. All it can do is change the shapes and sizes of the parts (e.g. canoe body and keel) that the boat already has. We would like to study the process through which structural innovations such as winglets can be introduced.

Our idea, based on a reconstruction of the way winglets were originally devised, is the following: Both the physics of the artifact and additional constraints such as cost or the 12-Meter Rule introduce a tradeoff among the forces at work. A structural innovation, in essence, is a way to modify the tradeoffs so as to make a better one available. E.g., in our reconstruction of the discovery of winglets, a combination of the 12-Meter Rule (which limits keel depth) and physical reasoning (which calls for a low center of gravity for the keel) leads to a design with a keel that has a long lower edge. But this kind of design gives a very bad tradeoff between sideways lift provided by the keel and the drag such lift causes. It was recognized that a similar tradeoff problem had been faced in aircraft design, and had been solved by adding a structure analogous to winglets on the keel.

Thus, our idea is to analyze key (limiting) force tradeoffs involved in a given design, and then to generate a set of candidate tradeoffs that would be most useful to improve. This information would then be given to a human user of the system, in the hopes that the user will know of, or can invent, a structural modification that changes the limiting tradeoffs in the desired direction.[3] This process can be seen as starting by identifying/characterizing an impasse in a given design, relative to desired specifications, and looking for ways of removing the impasse.

---

[3]We believe it may be possible to automate this process of going from tradeoff to structural modification, perhaps using techniques like those of Sycara et al [2], but we do not plan to work on this in the immediate future.

4. FAD Research; Application to Microprocessor Design and to Machining Sequence Planning

In addition to the domain of boat design, we have also been working on a testbed system in the domain of microprocessor design, although at a lower level of effort. Current participants in this area include Drs. Steinberg and Ellman, and a graduate student, Chun Liew.

Microprocessor design, indeed digital circuit design in general, might be thought of as a domain where constraints are largely decomposable (e.g. see [9]). However, it turns out that this is true only if all we care about is *functional correctness*. If there are constraints on the *quality* of the design, i.e. its use of such global resources as power, area, and time, then we no longer have decomposable constraints. E.g., we do not necessarily get the smallest area for a microprocessor as a whole by designing the registers to take the smallest possible area - such a design for registers may force a very large-area design for the data path.

Microprocessor design is further complicated by the huge number of decisions that need to be made. This necessitates a design process that proceeds through *abstraction spaces*, making different kinds of decisions about the circuit, e.g. choosing how many of which kinds of functional units will be included such as multipliers and adders, allocating operations to time steps and to functional units, allocating intermediate values to physical registers, and allocation data flow paths to physical buses. Thus, throughout the design process we are searching in spaces of *partial* designs. The problem is that with a partial design we can only estimate its use of global resources. Indeed, existing High Level Synthesis systems search the various abstraction spaces by using such estimators to predict the ultimate effect on resource use of decisions within these spaces. However, in this domain there are many problem-specific interactions, and such estimators cannot be all that accurate. To get an accurate measure of resource use, we need to have a *complete, concrete* circuit.

Since we can only evaluate how a design meets its resource constraints if we have a complete design, it appears that we need a design process which searches in a space of complete designs, and not spaces of partial designs. Our approach to this task is to do just that, to search in the space of complete designs, but not to do so with operators that directly modify a design. Rather, our initial design is produced by the kind of search in abstraction spaces described above. Then each new candidate in the space of complete designs is created by *adding constraints* to one or more of the abstraction spaces (e.g. insisting that two operations not be scheduled for the same time step), and redoing the original design process but observing these new constraints. Thus we have an outer loop searching in a space of complete designs, and an inner loop searching in spaces of partial designs. Another way to look at this is as *iterative design*, with analysis at the end of one iteration providing *feedback* to earlier stages of the next iteration.

We have implemented a system called FAD (Feedback Aided Design) which embodies these ideas, and have been testing FAD. FAD works by using (manually derived) methods for detecting places in a design where the approximate evaluators have misled the search in some abstraction space, and for inferring constraints that will correct this error. (See previous reports or [3] for details.) We have been testing FAD by using it to control the iteration in three existing systems: ASP, MACHINIST, and most recently BUD/DAA.

ASP is a PROLOG based microprocessor synthesis system developed by our collaborator, Dr. Al Despain, from USC. We have designed several modifications and enhancements to the ASP system.

The modifications to ASP will provide it with additional capabilities of (1) completely automated control of design iterations, (2) reasoning about resource goals and (3) passing information from lower design levels to higher design levels so that the design will be more informed. We have completed the first of these enhancements and are currently working on the second and third phases. The FAD system is currently able to analyze the solutions generated by ASP and we are currently adding optimization operators to ASP, e.g., to select a given number of pipeline stages. There are some problems, however, because ASP itself is still under development by the USC team.

BUDD/DAA is an older, more stable system developed at AT&T Bell Labs by Ted Kowalski et al. Recently we have begun looking at it as an additional testbed for FAD. We have started by applying FAD to the control of search in one of BUD/DAA's abstraction spaces - a search for a good clustering of operations into functional units. This is a problem that Dr. Kowalski tells us has resisted his and others' efforts at improvement. Initial results indicate FAD's control improves the clustering by 15 - 20%, which Dr. Kowalski regards as substantial.

MACHINIST is a system for planning machine tool operations developed at CMU by Dr. Caroline Hayes (currently at the University of Illinois Urbana-Champaign). Like microprocessor design this task involves multiple, global resource constraints, and thus is also a natural task to try FAD on. We have completed the modifications and enhancements to FAD and MACHINIST that enable them to work together. We are currently performing experiments to determine how well the combined system performs.

The goal of these experiments is to determine the range of problems for which the FAD technique is appropriate, and the problem conditions that make FAD more useful. Also, the experiments provide a sanity check on the technique, because we will be testing it on design problem solvers that were not developed by us. Since the ASP, BUD/DAA, and MACHINIST systems were designed and written by three very different and separate groups of researchers (at USC, Bell Labs, and CMU) and the systems were implemented in three different programming paradigms and languages (PROLOG, C, and OPS5), they provide a very good testbed to evaluate our ideas.

4.1. Synthesis of Heuristics

In related core research, Dr. Tom Ellman has been investigating methods for automatic synthesis of search control heuristics from the formulation of a design problem. One portion of this work has been directed to the development of methods for constraint satisfaction problems (CSPs) of the type that are often found in design, including in the abstraction spaces involved in micro-processor design. In particular, he has developed techniques for automatically synthesizing two types of heuristics for CSPs: Filtering functions are used to remove portions of a search space from consideration. Evaluation functions are used to order the remaining choices. These techniques operate by first constructing exactly correct filters and evaluators. These operate by exhaustively searching an entire CSP problem space. Abstracting and decomposing transformations are then applied in order to make the filters and evaluators easier to compute. An abstracting transformation replaces the original CSP problem space with a smaller abstraction space. A decomposing transformation splits a single CSP problem space into two or more subspaces, ignoring any interactions between them. Both types of transformation potentially introduce errors into the initially exact filters and evaluators. The transformations thus implement a tradeoff between the cost of using filters and evaluators, and the accuracy of the heuristic advice they provide. These techniques have been shown capable of synthesizing useful heuristics in domains such as floorplanning and job-scheduling, among others. Ongoing research is aimed at applying and extending these techniques to a VLSI design task: Automatic synthesis of functions that estimate low level resource usage given only an abstract, high level circuit design. This research is part of **collaborative work with Dr. A. Despain (USC)**.

5. Other Core Work

This section will describe three parts of our core work that are less directly tied to our testbed domains and systems: automated formulation of approximate thermal models, conceptual/concurrent design, and qualitative reasoning.

5.1. Automated Formulation of Approximate Thermal Models

We are continuing our work on automating the formulation of mathematical models of physical systems. Main participants are Dr. Louis Steinberg and Ringo Ling. This work is being done in the domain of thermal systems. In such systems, as in the fluid flow domain, a full simulation of all known physical processes at work would be computationally intractable, so approximate models must be developed. Developing such models currently requires human judgment. Automating the process would be practically useful and hopefully would shed light on related problems of model formulation in qualitative physics and more general issues of problem formulation.

During last year, we have built a prototype system; it derives mathematical models of heat flow processes involving a single rectangular object. The input consists of three parts: the structural description of an object, the initial and boundary conditions of the object, and the query (e.g. to find the heat flow or the temperature distribution) and its spatial/temporary dependency (e.g., whether we want to know how the temperature varies across time and/or along the X axis). The output is a set of mathematical equations which describes the heat transfer behavior of that object. Various mathematical models can be generated depending on the object’s dimensions and the query. The models include algebraic equations, ordinary differential equations and partial differential equations. The system has three basic components: system boundary identification, instantiation of relevant energy processes and mathematical transformation. The system uses the spatial dependency of the query to choose a control volume, from which relevant energy processes are instantiated. Finally, the system translates the processes into their mathematical representation, and carries out simple transformations and simplifications.

How to choose the right kinds of control volumes, how to handle the interaction of energy processes due to structural connections, and how to prune away irrelevant energy processes are some of the important research issues we have identified in building the prototype. We are planning to extend the prototype for objects with composite structures to address the first two issues, and we are incorporating order of magnitude reasoning to address the last issue of pruning away irrelevant processes.

5.2. Conceptual/Concurrent Design

Over the past year Prof. Tong continued to develop a model of design that integrates design with reasoning about manufacturing, usability, etc. He and Andrew Gomory implemented a computer environment for designing appliances that can be viewed as assemblies of catalog parts and connectors (e.g., an electric juicer or a desk lamp). The design environment is intended to facilitate iterative conceptual redesign. The long-term goal for this environment is that it will enable the user to specify redesigns in three different modes: (1) structurally, by indicating a series of structural modifications to be performed to the current design; (2) behaviorally, by indicating a modification in the behavior of the design; and (3) criterially, by indicating a design criterion (e.g., usability or manufacturability) with respect to which the design should be improved. Our research prototype implements the first mode. If any of the specified structural modifications (e.g., attach the shaft of a motor to the bottom of the "reamer" of the manual juicer) cannot be immediately accomplished because some precondition is not satisfied (e.g., the motor shaft and the reamer bottom are not physically adjacent), the system tries to achieve these preconditions (e.g., by drilling a hole in the base of the manual juicer).

5.3. Qualitative Reasoning

Additional core research on modeling physical systems is being pursued by Drs. Tom Ellman and Haym Hirsh, and a graduate student, Ke-Thia Yao. This work is aimed at overcoming some well-known problems with existing methods for qualitative simulation (QSIM) about physical systems. Existing QSIM systems are unable to make precise predictions, because they abstract away too much information about the physical system under consideration. Our research is intended to overcome this problem by combining QSIM with numeric simulation and machine learning. In particular, we are attempting to use inductive learning techniques to find rules for resolving the ambiguous predictions made by QSIM. Our initial efforts have developed a set of tools needed to carry out experiments in this area. In particular, we have developed a system that is capable of resolving data from numeric simulations with less precise descriptions generated by qualitative simulation. We expect to carry out the inductive learning experiments during the coming year.

6. Dissemination, Planning and Infrastructure Development

Dr. Amarel presented a keynote address on ’Themes and Directions of AI Research; Opportunities and Challenges’ at the Ninth National Conference on AI (AAAI91), held in Anaheim, CA on July 14-19-1991; he organized and chaired a panel on AI and Design for IJCAI-91, held in Sydney, Australia on August 24-30, 1991; he presented a keynote address on ’AI and Computational Design’ at the International Symposium on AI, held in Cancun, Mexico, on November 13-15, 1991; he gave an invited address on ’Problem Solving in AI; Issues in Problem Representation’ at the ’Frontiers in Computing’ Symposium honoring Professor Bledsoe, held in Austin, Texas, November 15-16, 1991; he presented the keynote address on ’Issues of Representation and Control in Machine Learning within a Problem Solving Paradigm’ at the Multistrategy Learning Workshop, held at Harper’s Ferry, VA, on Nov 7-9, 1991; he gave an invited seminar on ’AI and Computational Design’ at ETS, Princeton, N.J., on Dec 16, 1991; he will present a seminar on ’AI and Design; the CAP project at Rutgers’ at the Engineering Research Center, CMU, Pittsburgh, PA, on April 17, 1992; he will present a keynote address on ’A perspective on problem representation and reformulation in AI’ in the Workshop on Change of Representation and Reformulation, to be held in Asilomar, CA, on April 28-30, 1992; and he will give an invited address on ’AI and Problems of Design’ at the 25th Celebratory Symposium of the Dept of CS at SUNY Buffalo, on May 4, 1992. Dr. Amarel participated in the 1991 ISAT summer study on ’Intelligent Manufacturing’, which concentrated on information system requirements for overall enterprise integration, including engineering design and manufacturing. He continues to be guest editor for the IEEE/Expert special series of articles on AI and DARPA’s Strategic Computing Initiative.

Dr. Ellman is currently organizing the AAAI Workshop on Approximation and Abstraction of Computational Theories, to be held at the AAAI-92 conference in San Jose in July, 1992. The workshop will include investigators from a variety of areas including machine learning, qualitative physics, computer-aided design and mechanical engineering, among others. It is intended to facilitate an exchange of views between diverse research communities that share a common interest in automatically generating approximations or abstractions for tasks such as problem-solving, diagnosis, simulation and design, among others.

Dr. Gelsey presented a paper on "Using Intelligently Controlled Simulation to Predict a Machine’s Long-Term Behavior" at the Ninth National Conference on AI (AAAI91). He will present work on "Using Artificial Intelligence to Control Fluid Flow Computations" at the NASA Workshop on Software Systems for Surface Modeling and Grid Generation to be held April 28-30, 1992 at the NASA Langley Research Center.

Dr. Steinberg is a member of the Organizing Committee for the Second International Conference on Artificial Intelligence in Design to be held in Pittsburgh, Pennsylvania, in June 1992; he is a member of the Organizing Committee for the Workshop on Preliminary Stages of Engineering Analysis and Modeling to be held in conjunction with that conference; he was a member of the Organizing Committee for the IFIP conference on AI and Design in Columbus, Ohio, September, 1991; he was a member, of the Organizing Committee for the First International Conference on Artificial Intelligence in Design, Edinburgh, Scotland, June 1991; he was an invited speaker on a panel on AI and Design at IJCAI-91, Sydney, Australia, August, 1991; he gave an invited talk on "Combining Innovation and Perspiration" at the Workshop on Artificial Intelligence in Design held at IJCAI-91, Sydney, Australia, August, 1991; he gave a talk at the "How Things Work" workshop at AAAI-91, Annaheim, August, 1991; A paper of his titled "Design as Top-Down Refinement Plus Constraint Propagation", will appear in *Artificial Intelligence In Engineering Design*, edited by Dr. Tong and Dr. D. Sriram.

A paper by Liew, Steinberg, and Tong on "Use of Feedback to Control Redesign" appeared in the IFIP conference on AI and Design in September, 1991.

Dr. Tong was the Program Chair of the Scientific/Engineering Track (primary track of 3 tracks), IEEE Conference on Artificial Intelligence Applications, Miami, Florida, March 1991; he is a Program Committee member for the AAAI-92 Workshop on Automating Software Design: "Domain-specific software design systems", July 1992; he is a Program Committee member for the AAAI-sponsored "Workshop on Reformulation and Representation Change", May 1992; he was a Program Committee member for the AAAI-91 Workshop on Automating Software Design, August 1991. He is on the Advisory Board for the Seventh International Conference on Applications of Artificial Intelligence in Engineering, University of Waterloo, Ontario, Canada, July 1992 and the Advisory Board for the Second International Conference on Artificial Intelligence in Design, Carnegie Mellon University, Pittsburgh, PA, June 1992; he was on the Advisory Board for the Sixth International Conference on Applications of Artificial Intelligence in Engineering, Boston, Massachusetts, July 1991 and the Advisory Board for the First International Conference on Artificial Intelligence in Design, Edinburgh, Scotland, June 1991; he presented a tutorial on "Artificial Intelligence in Engineering Design," with D. Sriram (MIT) at the 1991 AAAI Conference, in August; he was an invited participant in the Workshop on Research Directions for AI in Design, UCLA, January 1992; he was chair of the "Testing and maintenance" session at the AAAI-91 Workshop on Automating Software Design, August 1991; he was chair of the "Engineering the Interaction of Software

Design Systems" panel at the AAAI-91 Workshop on Automating Software Design, August 1991; he has co-edited a three volume collection of papers, entitled "Artificial Intelligence in Engineering Design", which is being published by Academic Press, and will appear in the early summer of 1992; he wrote a chapter in a book entitled "Knowledge-Aided Design" (Academic Press, 1991) on "Eliminating Search in Design Via Learning"; he wrote a chapter in a book entitled "Automated Software Design" (AAAI Press, 1991) on "A divide-and-conquer approach to knowledge compilation."

7. Plans for Year 3

We will now outline our plans for the coming year. Note that with the no-cost extension we have requested, the "year" actually runs until July, 1993. First we will discuss our plans in our two system development efforts, and then our work on the core issues.

7.1. System Development in Ship Design

In the Ship Design Associate implementation work, we have two goals for Year 3:

- By July, 1992, demo and experiment with an improved Design Associate for sailboat design. This includes:
    - Implement automatic selection of an initial design, based on performance for the new problem conditions.
    - Implement an improved version of our methods for structuring the design space and for multi-level search of the space, and test this version experimentally to determine how the search time and the quality of the final design compare with those produced by searching without using this structuring.
    - Implement an improved version of our methods for intelligent selection of evaluation models and test it for speed of search and quality of results against search using only a single model and/or that use less informed methods to do select the model.
- By the end of the year to identify a task of practical significance to the Navy and implement a prototype Design Associate for this task based on our implementation of the Design Associate for sailboats. We plan to collaborate with Howard Chatterton and his colleagues at NAVSEA in this effort. Initial discussions to this end have already taken place.

7.2. System Development in Microprocessor Design

By the end of the year we plan to have fully integrated FAD with ASP and/or BUD/DAA, including methods to

- Completely automated control of design iterations
- Control design using using explicit reasoning about goals for usage of multiple global resources
- Pass information from later stages of one design iteration on to earlier stages of later iterations.

We also plan to have carried out experiments to test how much using FAD improves the designs produced by underlying systems, both for the microprocessor design task and for the machining-sequence plans produced by MACHINIST.

7.3. Core Research

In our core research areas, we plan to do a number of things by the end of Year 3. Several of these center around reasoning about tradeoffs between physical processes and forces such as the various drags and lifts that affect a sailboat. By the end of the year, we plan to implement and test our initial ideas about using such reasoning for the following tasks:

- Selecting an initial design
- Introducing structural innovations into the search space (with a human doing the actual retrieval of the innovation, based on guidance from the tradeoff reasoning)
- Conceptual design, where what is desired is information about what is possible and what the pros and cons of the alternatives are, rather than a specific design.

We plan to do this work in the context of the Design Associate for sailboats, although we hope to transfer it to the new Design Associate for the practical Navy problem as much as this is feasible.

In conjunction with the work on innovation, we plan to have carried out an initial study on how a system might support structural innovation by (semi-)automatically generating revised evaluation models that can be used to evaluate the performance of designs that include the innovative structure.

We also plan to pursue our ideas on using AI to support robust simulation. In the spatial reasoning area, we plan to to formulate the PMARC panelization problem as a search for appropriate transformations within a hierarchy of spatial representations for a yacht (or other physical object). In the modeling area, we plan to formalize the space of approximate physical models on which PMARC is based, in order to make all simplifying assumptions explicit so that PMARC output can be tested for consistency with these underlying assumptions. We plan to develop these ideas and, based on them, produce an initial implementation of a system that can use PMARC to automatically analyze a broader range of physical artifacts than can be achieved with current methods. (This is of course closely related to the problem of evaluating design with innovative structures.)

In the core work on synthesis of heuristics, we plan to have an initial implementation of methods to synthesize heuristics from experience that can be used to guide search in the domain of microprocessor design in a system such as ASP or BUD/DAA.

In addition, we plan to have done the following things by the end of the year:

- We plan to implement and test our ideas on geometry optimization using computational fluid dynamics.
- In the work on generating thermal models, we plan to have extended our implementation to handle artifacts with multiple components, and to have tested it on a number of problems provided by our domain experts.

I. Publications

The following is a list of publications produced by our group during the past year.

Amarel, S. ’Introduction to panel on AI and Design’, in Proc IJCAI-91, August 24-30, 1991, Sydney, Australia; Morgan Kaufmann Publishers, San Mateo, CA.

Amarel, S. ’AI research in DARPA’s Strategic Computing Initiative; Guest Editor’s Introduction to a series of articles on DARPA’s Strategic Computing Initiative’ IEEE/EXPERT, Vol 6, No. 3, June 1991; issued as Tech Report, CAP-TR-4, LCSR, Rutgers University, New Brunswick, N.J., September 1991.

Gelsey, A., ’Using Intelligently Controlled Simulation to Predict a Machine’s Long-Term Behavior’, in Proceedings, Ninth National Conference on Artificial Intelligence, August 24-30, 1991, Sydney, Australia; Morgan Kaufmann Publishers, San Mateo, CA.

Gelsey, A., ’Using Artificial Intelligence to Control Fluid Flow Computations’, in Proceedings, NASA Workshop on Software Systems for Surface Modeling and Grid Generation, April 28-30, 1992, NASA Langley Research Center.

Liew, C.W., Steinberg, L., and Tong, C.H., ’Use Of Feedback To Control Redesign’, in Proceedings of the IFIP TC5/WG5.2 Working Conference on Intelligent Computer Aided Design, Columbus, Ohio, September, 1991. Also available as Report CAP-TR-3, Lab for Computer Science Research, Rutgers University, New Brunswick, NJ.

Steinberg, L., ’Combining Innovation And Perspiration’, in Preprints of the Workshop on Artificial Intelligence in Design, IJCAI91, August 24-30, 1991, Sydney, Australia. Also available as Report CAP-TR-2, Lab for Computer Science Research, Rutgers University, New Brunswick, NJ.

Will P., Uncapher, K. (eds) ’Intelligent Manufacturing’ (with participation by S. Amarel, A. Despain, J. S. Brown, E. Feigenbaum, S. Helland, C. Holloway, J. Lardner, F. Prinz, A. Reisman, L. Rowe, R. Shorey, D. Sieworek, R. Smith, B. Spinrad, B. Winner, J. Yusco), report of ISAT summer study, August 1991, IDA, Alexandria, VA.

References

}

[1] Saul Amarel.
Problem Solving.
Encyclopedia of Computer Science and Technology.
Wiley, 1992.
Also available as Report CAP-TR-5, Lab for Computer Science Research, Rutgers University, New Brunswick, NJ.

[2] K. Sycara and D. Navinchandra.
Index Transformation Techniques for Facilitating Creative Use of Multiple Cases.
In *Proceedings of the 12th International Joint Conference on Intelligence*, pages 347-352. Sydney, Australia, 1991.

[3] C.W. Liew and L.I. Steinberg and C.H. Tong.
Use Of Feedback To Control Redesign.
In D.C. Brown and M. Waldron and H. Yoshikawa (editor), *Proceedings of the IFIP TC5/WG5.2 Working Conference on Intelligent Computer Aided Design*. 1991.
Also available as Report CAP-TR-3, Lab for Computer Science Research, Rutgers University, New Brunswick, NJ.

[4] J. McDermott and L. Steinberg.
AI and Design: Course Notes from the IJCAI 89 Tutorial.
August, 1989.
Also available as Rutgers AI/Design Project Working Paper Number 132.

[5] M. Orelup, J. Dixon, P. Cohen, and M. Simmons.
Dominic II: Meta-level Control in Iterative Design.
In *Proceedings of the Seventh National Conference on Artificial Intelligence*. August, 1988.

[6] Ramachandran, N., Shah, A., and Langrana, N.
Expert System Approach in Design of Mechanical Components.
Technical Report CAIP-TR-058, Center for Computer Aids for Industrial Productivity, Rutgers University, January, 1988.
Proceedings of the 1988 Computers in Engineering, July 31 - August 3, 1988, San Francisco, CA. Also Rutgers AI/Design Project working paper 88.

[7] N. Ramachandran and N. Langrana and L. Steinberg.
Initial Design Strategies for Iterative Design.
In *Proceedings of 2nd International Conference on Design Theory and Methodology*. ASME, Chicago, Ill., September, 1990.
Rutgers AI/Design Group Working Paper 159.

[8] L. Steinberg.
Dimensions for Categorizing Design Tasks.
In *AAAI Spring 1989 Symposium on AI and Manufacturing*. March, 1989.
Available as Rutgers AI/Design Project Working Paper Number 127.

[9] Steinberg, Louis.
Design as Top-Down Refinement Plus Constraint Propagation.
In C. Tong and D. Sriram, *Artificial Intelligence In Engineering Design.*
Academic Press, New York, 1991, Chapter 8.

Table of Contents