Methodology Discovery for Multi-disciplinary Design Problems

Cirrus Shakeri

Worcester Polytechnic Institute, 1998

Supervisors: David C. Brown & Mohammad N. Noori

Introduction and Aims of the Research

Introduction

The scope of this research is the multi-disciplinary design of engineered systems. The research aims to develop and explore a new approach *to discovering design methodologies for multi-disciplinary design problems*. The objective is to demonstrate the generation of specific design methodologies using that approach.

By design methodology we mean a specific scheme for organizing reasoning steps and domain knowledge to construct a solution for a particular problem in a particular domain. We are not referring to a general-purpose methodology for design. It provides both a conceptual framework for organizing design knowledge and a strategy for applying that knowledge [Sobolewski 1996]. It can provide the knowledge for decomposing the problem into sub-problems, synthesizing partial designs, evaluating and then combining them into partial designs, ordering design tasks, and discovering and resolving conflicts.

Current multi-disciplinary design methodologies are based on *ad-hoc* strategies for handling the complexities that multiple points-of-view bring to the design process. They reduce the complexity but give up the potential advantages of diversity; are based on compromising between different disciplines rather than collaborating between them; and are not as systematic, efficient and effective as they could be.

Most common methodologies use sequential design to overcome the complexities of multi-disciplinary design. Hence, information sharing between different disciplines is limited to the interfaces between disciplines [Levitt et al 1991], and conflicts between disciplines are not discovered until they are very expensive to resolve.

There is often a lead discipline in sequential design, perhaps a designer that makes key decisions, anticipating and removing conflicts. Other disciplines conform to those decisions, perhaps preventing them from producing their best solutions. In a lead-discipline approach a single point-of-view dominates, favoring constraints from that discipline. This produces a lower quality design product and increases the number of iterations required to reach an answer. Hence a key goal for better methodologies, used in this research, is to ensure *well integrated* reasoning that provide equal opportunity to all the disciplines involved.

The approach to generating better multi-disciplinary design methodologies was computer *simulation* of the design process. Analyzing the behavior of physical systems in engineering applications by computer simulation using mathematical models has been a powerful tool in engineering, particularly in situations preventing experiments with physical prototypes. This work extends the use of simulation-driven analysis to engineering design research, by applying it to the design *process* instead of the design product. Here “real experiments” would be extremely costly.

A *computational model* was developed and implemented to simulate the multi-disciplinary design process. By simulating under many different conditions, and analyzing the performance, detailed understanding of the design process is gained [Shakeri et al 1998]. As for simulations of physical systems, the computational model used is a *simplified* one in which the design activities that are usually carried out by humans are performed by software agents in a simplified manner. This research used the multi-disciplinary domain of robot arm design [Rivin 1988].

Importance

Using system-developed methodologies allows effective and efficient practices to be used from the *start* of a project instead of being learned from experience. These new methodologies are radically different from the sequential, discipline-based ones. Integration reduces the number of failures and backtracking by facilitating information sharing, thus saving resources and reducing design time. Integration also provides collaboration between different participants that, as a result, enhances the quality of the design.

The agent-based approach that we have adopted for building the computational model allows the incorporation of new technologies systematically and quickly through the addition or deletion of agents [Brown et al 1996] [Wooldridge 1997]. Thus new knowledge can be added, and old knowledge removed rapidly. Running a modified system will result in new designing behaviors being simulated, allowing production of new methodologies in response to a change in knowledge. In addition, design processes can be biased toward more environmentally friendly products, by altering the preferences for the alternative design methods that are built into each agent.

In industry, the number of specialists is increasing, while the number of generalists, capable of doing system integration, is decreasing. An increasingly specialized technological environment tends to force designers to concentrate on some disciplines more than others. Also the knowledge burden on the designer keeps increasing due to more materials and more options [National Science Foundation 1996]. Thus it is becoming harder to develop methodologies for the integration of multiple disciplines in design. This research directly attacks this problem.

Computers have mostly been used to support the manipulation and analysis of design product information. This work focuses on the design process, an aspect that has not benefited from computers very much. Simulation of design processes based on a multi-agent paradigm is a new area of research that has a high potential for practical as well as theoretical impact on the design of products. The use of multi-agent systems technology is growing rapidly with the development of Java-based systems and agent access across the world-wide web.

The research is also important because it recognizes the importance of incorporating knowledge, judgement and experience. “System integration, many consider, is an ill-structured problem... No specific rules have to be followed when doing integration... Experienced designers deal with system integration using judgement and experience. Knowledge-based programming technology offers a methodology to tackle these ill-structured integration and design problems” [Sobolewski 1996].

According to NSF’s report on Research Opportunities in Engineering Design [National Science Foundation 1996], “research areas that will have greatest impact on engineering design over the next 10 years are: Collaborative Design Tools and Techniques, Prescriptive Models/Methods, System Integration Infrastructure/Tools, and Design Information Support Systems". This work covers all of these areas of research and hence is expected to have a strong impact.

Research Approach

What was Developed

Part of the goal of the research was to produce an "approach" (i.e., to producing methodologies). First we will describe in more detail the approach that was developed during the research. Then we will describe the way the research progressed: i.e., the approach to the approach!

This work proposed a new approach to the problem of producing better design methodologies for multi-disciplinary design based on the tight integration of different disciplines. The discipline-sequential approach, while poor, is quite simple. Its flaws are well known and have been part of the motivation for concurrent engineering [Brown et al 1996].

However, integration tends to make the design process more complicated. To overcome this complexity, a computer system was developed based on a multi-agent systems paradigm in order to automate the simulation of the design process. The system also allows multiple design problems to be simulated in a small amount of time.

The system simulates examples of multi-disciplinary design processes while applying *integration principles*. The principles were developed from an examination of the literature. They include common design knowledge representation schemes and common communication mechanisms; design knowledge sharing among participants; cooperative problem-solving strategies among participants; simultaneous design processes where possible; and mechanisms for conflict discovery and resolution. The principles are embodied in the system both in its architecture and at run-time.

The large chunks of discipline-specific knowledge are broken into small pieces, typically at the level of a single design decision, and each piece is represented in the design system by an agent. Agent activation is triggered in an opportunistic manner and is unaffected by discipline boundaries. Agents might participate sequentially or *in parallel*. This leads to well-mixed use of knowledge from different disciplines, and the possibility of parallel design activity for tighter integration and better efficiency.

The multi-agent design system is run with a very large number of different design problems. This is done by systematically varying the individual design requirements across their ranges in order to cover the space of requirements. Hundreds of design problems are presented to the system. Some problems do not lead to a successful design.

For each problem the traces of the agent activations (i.e., knowledge use) during the course of the design process are recorded. The many recorded traces consist of orderly patterns of different design actions that have led to a design solution.

Candidate design methodologies are extracted by generalizing the patterns in the recorded design traces using clustering techniques. This both groups and identifies common aspects of related traces. The best clusters are the most 'convincing' methodologies. For each cluster identified, the commonalities in the Requirements are identified. This allows combinations of Requirements to be recognized as being most appropriately handled by a particular methodology.

Research Questions and Hypotheses

The *main hypothesis* for this research was that methodologies could be generated by using a computer to build up "experience" by simulating design activity. As the system built was indeed able to generate methodologies, this hypothesis was confirmed.

The question was "how?". Clearly doing it with real people was impractical. This led to the idea of simulation. The need for integration led to the notion that any knowledge should have the potential to be applied opportunistically at any time, and that the knowledge should be split into pieces. These ideas, and the analogy with the design teams used to support Concurrent Engineering, made us decide to use a multi-agent design system.

The multi-agent approach intuitively captures the concept of deep, modular expertise that is at the heart of knowledge-based design [Lander 1997]. A multi-agent system is composed of multiple interacting agents, where each agent is a coarse-grained computational system. Agents are used as an abstraction tool for conceptualizing, designing, and implementing the knowledge-based design approach. An agent is a self-contained problem solving system capable of autonomous, reactive, pro-active, social behavior. It is a powerful abstraction tool for managing the complexity of software systems [Wooldridge 1997]. Thus, the multi-agent paradigm not only matched the problem, but also provided some Software Engineering advantages.

We started by investigating "methods", "methodologies" and "integration", as well as studying the literature on multi-disciplinary design. The latter confirmed the belief that there was a need for a systematic way of building good methodologies, and that many of those currently in place were ad hoc.

Early in the research we picked a domain in which to work: one that was well known to the student, had a clear multi-disciplinary flavor, was of a manageable size without being trivial, and appeared to have no strong, existing methodology of the type we were seeking. Robot arm design seemed to be perfect. It demonstrated well the tendency for researchers from each discipline to write about the design problem as if their discipline's contribution were dominant.

We started by implementing a "base level" working system that used a non-integrated approach to robot arm design. This tested our understanding of the relevant knowledge and methods used. Early prototype implementation is an important research technique that enhances domain understanding, acts as a catalyst for learning programming and system development techniques, and also forces precise definitions of concepts.

The base level system also provided some feedback about where errors occurred during designing and where the knowledge might be decomposed into pieces. The choice of Java for the implementation allowed portability between systems and provided the ability to effectively handle agents in parallel.

Next the framework for a multi-agent design system, to be called Robot Designer (RD), was developed. The question of how to split the knowledge from each discipline into pieces was addressed, and the resulting pieces were encoded as agents, and added to the framework. Decisions were made as to what needed to be stored as a record of every agent's action, such that these traces might be able to form suitable methodologies. The traces were accumulated, but not the designs.

Much time was spent on the failure handling system for RD, so that the constraint failures could be recovered from, while allowing parallel paths through the agents to be recorded correctly.

An important issue was the relationship between design quality and methodology quality. Clearly one would like the methodologies produced to lead to high quality designs. The traces that actually lead to designs must include no failing constraints, and hence at least possess a certain level of quality. But they may not all be of equal quality. In addition, our simulation of the design doesn't include *all* of the design knowledge available, and hence may not lead to the best possible designs -- in fact, it would be foolish to pretend that one could make a perfect simulation of the design activity in a complex multidisciplinary situation. However, as we expected these methodologies to be followed principally by people, they only need to act as guidance, and the human designer should be able to ensure the quality of the result.

The compound hypothesis was that less precise knowledge might lead to adequate designs, *and* that adequate designs were associated with traces that could be used to form a methodology, *and* that this methodology was capable of guiding the production of high quality designs. This hypothesis was never *fully* explored.

A key question to be addressed at this point was how to exercise RD such that the whole design space was explored. This was important because we wanted to generate methodologies for all types of designs in the space. The approach taken was to drive the system with as many different sets of requirements as possible, such that the whole design space was explored. The hypothesis was that all reachable regions of the design space could be found by systematically varying the requirements in order to adequately scan the requirements space. By experimenting with the degree of change between requirements we were able to convince ourselves that this approach was successful.

Another issue that this raised, and which we spent a lot of time investigating, was the relationship between the requirements space, the design space and the trace space. These relationships were explored and revealed using graphical representations.

The next question to be addressed was how to form traces into methodologies. Several techniques were considered, and a clustering algorithm was selected. While other methods might be appropriate we focussed on just one. Once clusters were identified they were re-expressed as rules.

A final issue that was examined was how to best express the requirements that correspond to a particular methodology.

The main hypothesis, that methodologies could be generated using the computer by simulating enough design experience, was demonstrated by this research.

Results & their Evaluation

A knowledge-based model of design was adopted in order to implement the proposed strategies for integration. To implement the proposed model a knowledge-based multi-agent design system, RD, was developed that simulates the design process.

Both the general multi-agent design system architecture *and* the RD system developed are results of the research. The approach to breaking knowledge up into pieces such that they can be incorporated into agents is also a result from this research. Although methodologies were generated, and are also a result of this research, they were less important for this thesis.

The Java-based program, RD, was implemented for parametric design of a two degrees of freedom (2-DOF) planar robot arm. We used RD to solve a set of 960 design projects. Figure 1 shows how many projects followed a specific trace. The promising result is that many projects followed similar traces. The total number of possible traces is the product of the number of design approaches of all the designer agents. For the experiments shown in Figure 1 the total number of possible traces is 2,304. However, despite all those possible traces only 84 were followed to generate successful designs, i.e., less than 4%. One trace was common between projects that failed and projects that succeeded. Dashed lines show unsuccessful designs. At least 25% of attempts failed.

The low percentage of successful, relative to "possible", traces indicates that for each group of projects that followed a particular trace there is a unique combination of approaches leading to successful designs. Hence there is a high chance that if similar projects follow the same trace they will succeed in generating a successful design. As a result, the path followed by those projects can lead us to formulating a design methodology for the projects that followed that trace as well as projects that are similar.

The traces in the set of successful traces that are close enough can be clustered together to form a generalized trace. A generalized trace covers all the projects that followed each of the traces incorporated in the generalized trace. Design methodologies are formulated by extracting the correlation between a generalized trace and the design projects (i.e., the sets of requirements) that produced that trace. The sample design methodology that is shown below is the English translation of the correlation between design projects and the corresponding traces.

The phrase "do the design" refers to calculations or actions that can be done as a consequence of the key decision, and which generate values for some of the design parameters. In this example, choosing the structural safety factor allows designer to calculate the dimensions of the cross section of the robot's arm (e.g., diameter, thickness).

Methodology:

- choose the location of the base of the robot: *"left or below midway of the workspace length"*
- choose the material: *"steel stainless AISI 302 annealed"*
- select the shape of the cross section of the link: *"hollow round"*
- choose the structural safety factor: *"3"*
    - **do** the design and proceed to the next step
- choose the link 2 to link 1 length ratio: *"0.5"*
    - **do** the design and proceed to the next step
- pick the configuration of the arm: *"left-handed"*
- select the ratio of the cross section dimension of the link to minimum required by stress analysis: *"4"—if it fails select "3"*
    - **do** the design and proceed to the next step
- find the accessible region: *use Equation 2-4*
- find the deflection of the tip: *use Equation 2-14*
- choose the type of controller: *"PD"*
    - **do** the design and finish the process.

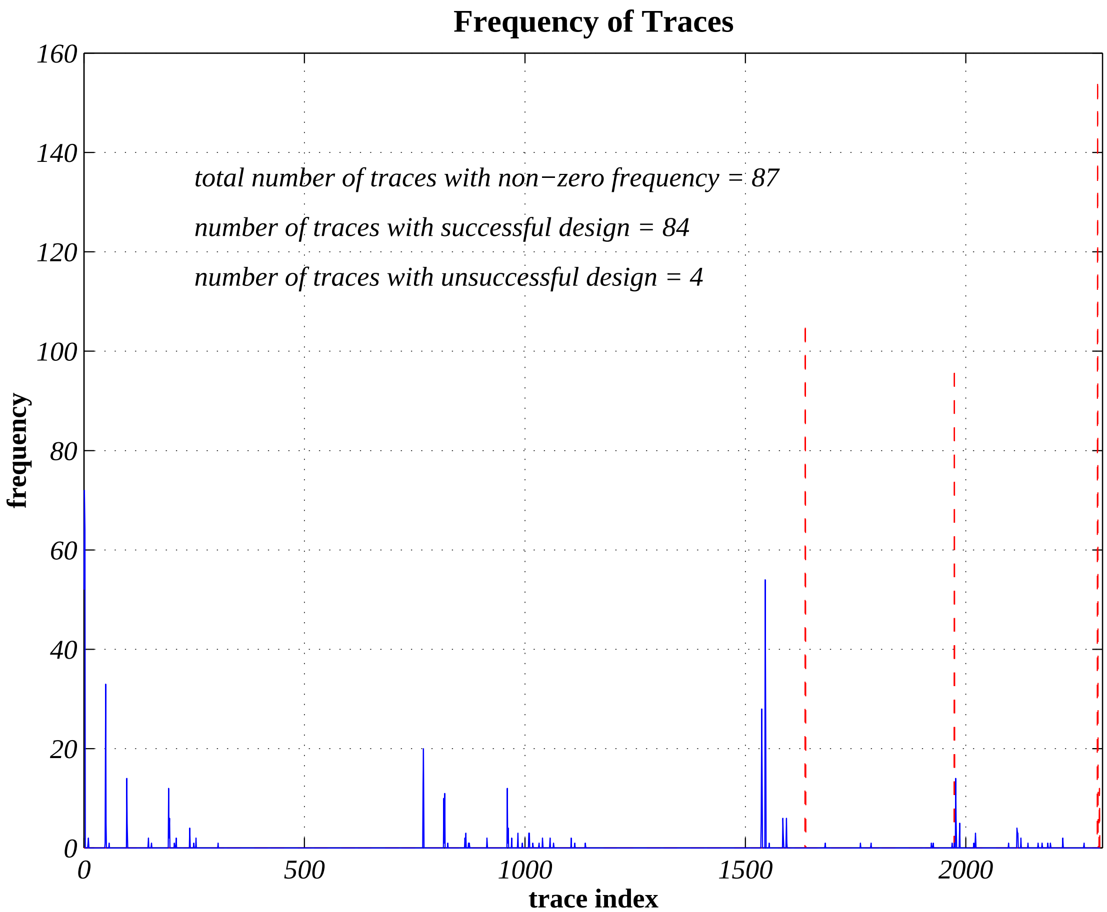

Figure 1. Frequency of Traces.

Conclusions about the Research Approach

The main characteristic of this research approach was the use of a computational model to simulate design activity, that, due to its size and complexity, would otherwise be impossible due to the time and cooperation required. The main positive feature of this approach is therefore that it allows the normally "impossible" to be handled. Other positive features should be self-evident from what has been presented so far.

As with all research, there are a variety of trade-offs and assumptions incorporated. Some of the assumptions need to be verified more rigorously.

The largest issue is that for every domain the approach requires collection of a very large body of knowledge -- recognized as a very difficult task -- and the construction and testing of a large and complex software system. Once that's done the methodologies should follow fairly easily, assuming that other domains behave in a similar way to our test domain.

However, there is a very large investment in time and effort required in order to start getting results. This approach will only be viable if the methodologies generated are extensively used and if they provide large gains in quality or cost/time savings. Currently, we have only limited evidence that such gains are in fact provided.

Also a big assumption is that both the simplified model of designing (e.g., with respect to failures) and the simplified knowledge that are embodied in the system, are not so simplified that the traces generated are atypical. Also assumed is that all traces that lead to successful designs should be included in the process that leads to methodologies. This is without regard to the quality of the designs that these traces produced: all we know is that they are ‘correct’. This may affect the quality of designs that can be achieved by following the methodologies.

Clearly this leads to a trade-off between the effort required to build the system (more complexity and authenticity leads to more system building effort) and the quality of designs that can be achieved by following the methodologies. This trade-off needs to be explored.

The interesting issue of the relationship between the clustering of designs, the clustering of requirements, and the clustering of traces (to form methodologies) still needs more exploration. The approach of systematically scanning across the requirements space needs to be investigated further to test for sensitivity to different domains and problem areas. Perhaps an adaptive scanning approach could be tried?

The final weakness is that it is very hard to test the methodologies that are generated under realistic situations, as this would require extensive use by multi-disciplinary design teams with many design problems. As a consequence it’s hard to establish whether the designs produced by using the methodologies are indeed of high quality.

Continuation of Project

The potential applications of this research are in multi-disciplinary design situations, such as those that occur throughout the automotive industries, where large gains can be achieved with integrated methodologies. In addition, current methodologies can be analyzed for flaws and bottlenecks, and necessary refinements made. New methodologies can be customized so that they are biased toward specific objectives such as manufacturability or being environmentally friendly. By applying this approach the response time for the incorporation of new technologies in design processes should be reduced. Methodologies can be refined as soon as a change occurs in the market or in the organization of the company.

At this time no additional work has been carried out on this project since the Ph.D. [Shakeri 1998] was completed. As with any large, student-driven software project, transferability is an issue. There would be a very steep and significant learning curve associated with taking over the complex Java code that was designed. Another serious issue is that finding a student with the right combination of CS and ME skills is very rare, and would take some time to develop.

We would very much like to experiment with confirming the quality and utility of the methodologies generated, especially for new multi-disciplinary problems. Automobile, Aircraft, Computer and Mechatronics design should be fruitful areas to investigate. The approach that we have proposed has been developed based on parametric design problems. Applicability of the approach to other types of problems needs to be investigated.

This research has proven that the following hypothesis is true: Computers can provide us with better ways of doing design by discovering design methodologies that integrate multiple disciplines into the design process. It has been shown that it is possible to use computers to simulate the design process, and then analyze the results of the simulation to synthesize design methodologies that have superior features. This forms the basis of a new approach to the study of multi-disciplinary design processes.

References

Brown, D. C., S. E. Lander & C. J. Petrie, “The Application of Multi-agent Systems to Concurrent Engineering”, Editorial: *Concurrent Engineering: Research and Applications,* Technomic Publ., Vol. 4, No. 1, March 1996, pp. 2-5.

Lander, S., “Issues in Multi-agent Design Systems”, *IEEE Expert: Intelligent Systems and their Applications,* Vol. 12, No. 2, March-April 1997, pp. 18-26.

Levitt, R. E., Y. Jin & C. L. Dym, “Knowledge-Based Support for Management of Concurrent, Multidisciplinary Design”, *AI EDAM,* Academic Press Ltd., Vol. 5, No. 2, 1991, pp. 77-95.

National Science Foundation, *Research Opportunities in Engineering Design,* NSF Workshop Final Report, April 1996.

Rivin, E. I., *Mechanical Design of Robots,* McGraw-Hill Book Company, 1988.

Shakeri, C., *Discovery of Design Methodologies for the Integration of Multi-disciplinary Design Problems*, Ph.D. Thesis, Mechanical Engineering Dept., WPI, Worcester, MA 01609, USA, December 1998.

Shakeri, C., D. C. Brown & M. N. Noori, “Discovering Methodologies for Integrated Product Design”, *Proceedings of 1998 AI & Manufacturing Workshop: State of the Art and State of Practice,* Aug. 31-Sept. 2, 1998, Albuquerque, NM, pp. 169-176.

Sobolewski, M., “Multiagent Knowledge-Based Environment for Concurrent Engineering Applications”, *Concurrent Engineering: Research and Applications*, Technomic Publ., V.4, N.1, March 1996, pp. 89-97.

Wooldridge, M., “Agent-based software engineering”, *IEEE Transactions on Software Engineering,* Vol. 144, No. 1, 1997, pp. 26-37.