

A WEB-BASED MONITORING SYSTEM FOR MULTIDISCIPLINARY DESIGN PROJECTS

James L. Rogers^{*}, Andrea O. Salas[†], and Robert P. Weston[‡]
NASA Langley Research Center

ABSTRACT

In today's competitive environment, both industry and government agencies are under pressure to reduce the time and cost of multidisciplinary design projects. New tools have been introduced to assist in this process by facilitating the integration of and communication among diverse disciplinary codes. One such tool, a framework for multidisciplinary computational environments, is defined as a hardware and software architecture that enables integration, execution, and communication among diverse disciplinary processes. An examination of current frameworks reveals weaknesses in various areas, such as sequencing, displaying, monitoring, and controlling the design process. The objective of this research is to explore how Web technology, integrated with an existing framework, can improve these areas of weakness. This paper describes a Web-based system that optimizes and controls the execution sequence of design processes; and monitors the project status and results. The three-stage evolution of the system with increasingly complex problems demonstrates the feasibility of this approach.

^{*} Senior Computer Scientist, FMAD, MDOB

[†] Research Scientist, FMAD, MDOB

[‡] Senior Research Engineer, FMAD, MDOB

Copyright ©1998 by the American Institute of Aeronautics and Astronautics, Inc. No copyright is asserted in the United States under Title 17, U.S. Code. The U.S. Government has a royalty-free license to exercise all rights under the copyright claimed herein for Governmental purposes. All other rights are reserved by the Copyright Owner.

INTRODUCTION

A multidisciplinary design project involves the integration of various discipline codes that may have been developed on different computer architectures and that must be executed on heterogeneous computers distributed over a network. New tools have been introduced to assist in this process by facilitating the integration of and communication among these diverse disciplinary codes. One such tool, a framework for multidisciplinary computational environments, is defined as a hardware and software architecture that enables integration, execution, and communication among diverse disciplinary processes. An examination of current frameworks¹⁻¹⁰ reveals weaknesses in various areas, such as sequencing, displaying, monitoring, and controlling the design process.

The World Wide Web is a tool for disseminating information to a widely distributed user base. Information is accessed through Web pages that are viewed with a Web browser. Web pages are becoming much more interactive, requiring new tools and techniques to exploit this capability. The purpose of this research is to explore how Web technology, integrated with an existing framework, can improve the weaknesses found in current frameworks

Web technology has much to offer in the monitoring and controlling of design projects. Web browsers offer convenient access to information residing on remote computers. Web software is available to most researchers, and many are already familiar with this interface. Because the Web promotes communication among codes and computers, it can be used to reduce design cycle time and costs, a critical

goal for both industry and government agencies.

For the current research, a Web-based system has been designed to assist in monitoring and controlling multidisciplinary design projects. A main project Web page must be developed with links to the process Web pages created by project team members. The process Web pages describe various processes of the project, such as discipline analyses and optimization. In addition, these Web pages may provide further links to information (e.g., data, documentation) which is associated with individual processes.

After all processes have been defined for a design project, the project team members identify the couplings among the processes. The sequence for executing these processes can then be derived from knowledge regarding the processes and their couplings. Note that in most multidisciplinary design projects, the design cycle involves iterations that must be taken into account in sequencing and executing the processes.

The project Web page monitors the status of the design cycle. This Web page contains links to process control options. Users can select process control options, such as disabling a process or manually invoking a process, to alter the process execution sequence. A main control program collects control requests and uses them to identify the current set of processes that are ready for execution. The Web pages associated with particular processes have the capability to provide additional monitoring capabilities.

This paper contains a brief discussion of the current framework development efforts, particularly in the area of execution sequence of design processes, followed by a description of sequencing and display tools. The evolution of the Web-based system is then described in detail, and the feasibility of this system is demonstrated with three increasingly complex design problems. The paper concludes with comments on the current research.

EXECUTION CONTROL IN MULTIDISCIPLINARY FRAMEWORKS

A great deal of effort is being expended in the U.S. and abroad by government agencies, industry, and universities in the development of multidisciplinary frameworks¹¹. In these frameworks, process execution can be handled in several ways:

(a) A master program written in a compiled language, such as C or C++, can be used to code a fixed execution sequence of processes. Processes may be executed through subroutine or system calls.

(b) A master program written in an interpreted or script language such as tcl, can also be used to code a fixed execution sequence of processes.

(c) Graphical methods can be used to link the processes and to provide more flexibility in the sequence of their execution. The underlying code actually establishes the links.

Before beginning this project, the authors determined through personal communication, that most framework developers are exploring methods for incorporating Web technology into their framework development efforts. However, these efforts had not been discussed in the literature at that time.

The sequence for process execution is typically predetermined and coded in current frameworks. No tools for optimizing the execution sequence or for deleting a process from the sequence if it is no longer contributing useful data are applied. Once the sequence is determined, the framework executes the processes sequentially or sometimes in parallel. This approach is usually referred to as process flow rather than data flow.

The system presented here is developed around a knowledge-based system made up of a problem-independent set of rules that determine the processes ready for execution and a problem-dependent set of facts that describe the individual processes and their couplings. The processes are ordered in an

optimum sequence to minimize iterations and data required as input before it has been computed. Processes not contributing useful data can be temporarily or permanently deleted from the sequence. Thus, this system provides a dynamic and flexible method for executing the processes found in a multidisciplinary problem. The system is further enhanced by introducing Web technology, which provides the mechanisms for interactive alteration of sequencing control and execution monitoring. The system capabilities are coded to facilitate integration into any framework, regardless of the current sequencing and execution methods of the framework.

PROCESS SEQUENCING AND DISPLAY

After all of the processes and couplings in a multidisciplinary project are identified, the optimum sequence for executing the processes should be determined. One tool for determining this sequence is the Design Manager's Aid for Intelligent Decomposition, otherwise known as DeMAID.¹² DeMAID is a knowledge-based software tool to assist a project manager in making decisions that potentially can reduce the time and cost of a design cycle.

Feedforward couplings indicate data are computed by one process before being used by another. Feedback couplings indicate a process requires data input before the data are actually computed. Feedbacks also imply iterations and estimated initial data; such couplings should be eliminated where possible. DeMAID minimizes the number of feedback couplings and when certain feedback couplings cannot be eliminated, processes are grouped into iterative subcycles.

Once the iterative subcycles have been determined, DeMAID sequences the processes within each iterative subcycle with a genetic algorithm in order to minimize the time and cost of the design cycle.¹³ After the iterative subcycles and the optimum sequence for execution have been determined, DeMAID displays the sequence in a Design Structure Matrix (DSM) format, which was originally

formulated by Steward.¹⁴ A DSM from the heart dipole problem, one of the sample problems described later in the paper, is shown in Figure 1.

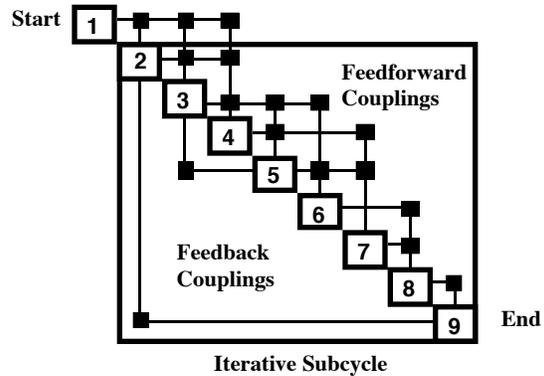


Figure 1. DSM for heart dipole problem.

The DSM in Figure 1 shows the design processes as numbered boxes on the diagonal. Execution of the processes starts at the top left of the matrix and proceeds to the bottom right. Output from a process is shown as a horizontal line that exits a numbered box, and input to a process is shown as a vertical line that enters a numbered box. The off-diagonal squares that connect the horizontal and vertical lines represent couplings between two processes. Squares in the upper triangle of the DSM represent feedforward couplings; squares in the lower triangle represent feedback couplings. In Figure 1, processes 2-9 are grouped into an iterative subcycle. The primary advantage of the DSM format over earlier display tools such as Program Evaluation Review Technique (PERT) charts, is the capability to group and display these iterative subcycles, which are common to design projects.

In the new Web-based monitoring system, the project DSM can be viewed with a Web browser. A Hyper Text Markup Language (HTML)⁵ file is used to describe the DSM. The DSM in Figure 1 is displayed in HTML format in Figure 2. A <META> tag in the HTML file automatically updates the DSM Web page every few seconds and uses different colors to indicate the current status of each process and coupling.

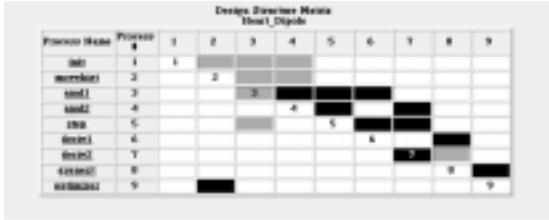


Figure 2. DSM for heart dipole in HTML format.

The HTML DSM in Figure 2 is a snapshot during an execution of the sample problem. The original HTML file uses colors to display the status of the project. This figure, using grayscale, reveals the following: along the diagonal, processes 2, 4, 5, 6, 8, and 9 are white and not ready for execution because they are waiting for data; process 3 is gray and is ready for execution; and process 7 is black because it has been deactivated. The colors of the off-diagonal cells reveal that only process 3 has all data available, that is, all coupling cells in column 3 of the matrix are gray. Coupling cells where data are not available are black. These colors change as different processes are executed. The output couplings remain gray for processes that are deactivated to indicate that the data for these processes are always available. The list of process names are links to Web pages describing each process in detail.

INITIAL WEB-BASED MONITORING SYSTEM

The initial Web-based system only provided the end user with a method for monitoring the processes that make up a multidisciplinary design project (Figure 3).

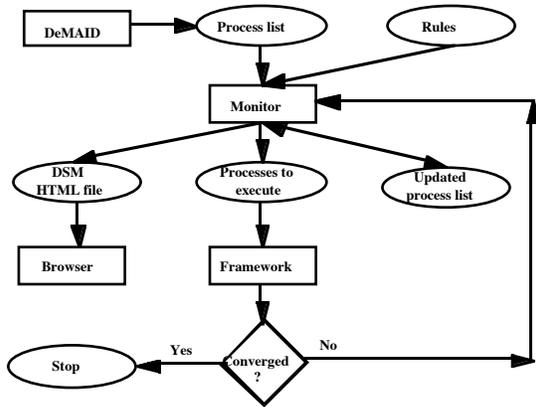


Figure 3. Initial monitoring system.

For this system, input to the monitor program consists of a file, output by DeMAID that contains a problem-dependent list of sequenced processes and links to the process Web pages, and a file with a set of problem-independent rules for determining which processes are ready for execution based on the satisfaction of their input requirements. The monitor program, written in C, calls the C Language Integrated Production System (CLIPS) knowledge-based system¹⁶ to produce three output files: an updated list of processes with their current status, a file listing those processes are ready for execution, and a HTML file for creating a DSM indicating the latest status of the design project that can be viewed with a browser. No control options were available in the initial implementation.

While the concepts used here are applicable to any multidisciplinary framework, a sample framework is written to test this implementation. This framework reads the file that contains the list of those processes that are ready for execution (which is output by the monitor program) and executes these processes sequentially. The first pass through the system assumes that all data that are represented by feedback couplings are available.

A sample problem¹⁷ with 21 processes (the DSM is shown in Figure 4), where each process consists of a single function, was used to test the initial system. This test was to insure that processes would be called at the correct time and that the monitoring system would correctly display the status of the problem. The functions consist of one or more input variables and compute a single output variable. The system was tested by executing the monitor program. The monitor program called the knowledge base to determine which processes had all their input data available, then passed the process names to the framework for execution. A Web browser was used to monitor the status of the problem. The monitor program continued to execute until the convergence criterion was met. The solution was considered to have converged when all output variables changed no more than 5% in the last three calls to the framework.

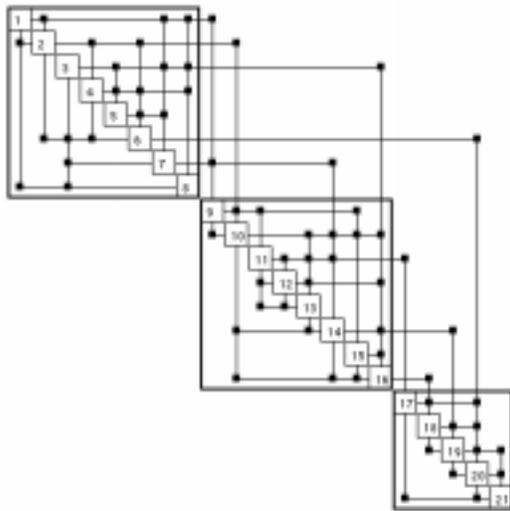


Figure 4. DSM for initial test problem.

**ENHANCED WEB-BASED
MONITORING AND CONTROL
SYSTEM**

The second implementation enhanced the monitoring capability of the Web-based system; and a method for controlling the processes that make up a multidisciplinary design project was added. The enhanced system is shown in Figure 5.

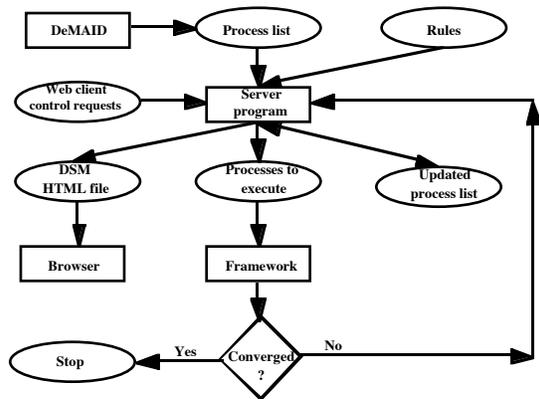


Figure 5. Enhanced monitoring and control system.

The heart dipole problem from the MDO Test Suite¹⁸ was selected to demonstrate the enhanced system. Although the heart dipole problem contains fewer processes than the initial test problem, the processes are more

complex in their function. The DSM for this problem is shown in Figure 1.

For this system, the main program that facilitates the monitoring and control is now referred to as the “server program.” Input to the server program is the same as the initial implementation with the addition of Web client control requests which allow the user to interactively activate and deactivate processes. The server program, a mixture of C and Java™ code,¹⁹ also uses CLIPS to produce the same three output files as the initial system. In addition to CLIPS, the server program uses various Java language features, such as applets, threads, sockets, and synchronized access to object functions to produce these files. A framework similar to the one used in the initial implementation is called from the server program to execute the processes. Details of this implementation can be found in Reference 20.

Java applets are implemented to display current and historical data and are invoked from within the HTML process pages. These applets can show the history of the objective function, selected responses, and design-variable values, as well as the status of the constraints. The applets reference the location of the current set of data to be processed. The locations are defined as parameters in the HTML pages.

Control options can also be set by using HTML forms. From the DSM, the user may click on a process name to link to its description as well as to display the current setting for the control option that needs to be modified. Changes can then be made and submitted interactively. Currently, the only available option is to deactivate or activate a process. For example, the user may want to deactivate a process if the output from the process is not changing. A deactivated process will not be called for execution even though all of its input data are available.

Two links may be chosen from the optimization process Web page, accessible from the main Web page for the project (Fig. 2). One link downloads a Web page that contains an applet for displaying the history of the objective function. The user can click the

applet Update button to update the objective function graph at any time. Alternatively, the user can specify that the applet automatically update the graph at regular intervals. The history of the objective function is displayed in a format similar to the one shown in Figure 6. Figure 6 displays the objective function for the heart dipole problem after nine iterations.

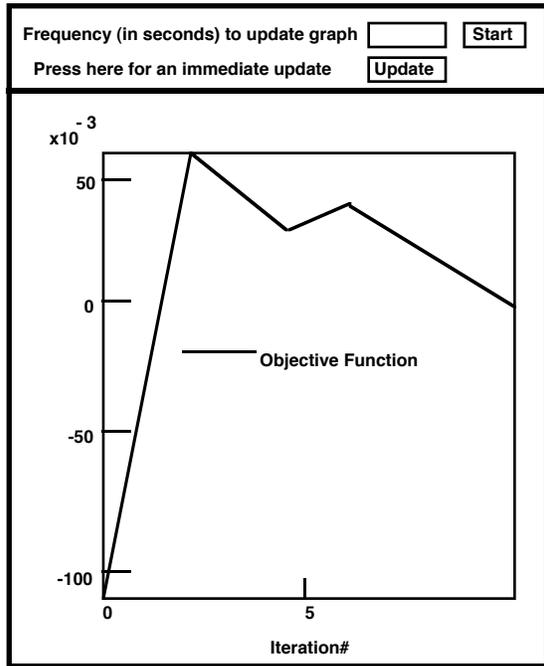


Figure 6. Objective function history from heart dipole problem.

The user can click on the second optimization Web-page link to download an HTML page that contains two applets: one for monitoring the design variables and one for monitoring the constraints. The design-variable monitoring applet displays the list of design variables. From this list, the user can select a design variable and view its history in a format similar to the one shown in Figure 7. A design variable was selected from the heart dipole problem after nine iterations to produce the history shown in Figure 7.

The heart dipole problem contains four constraints, for which current values are recorded after each iteration. The applet that monitors the constraint status displays the status of each constraint by using a color code: red indicates a violated constraint; green

indicates an active constraint; and yellow indicates a feasible, inactive constraint.

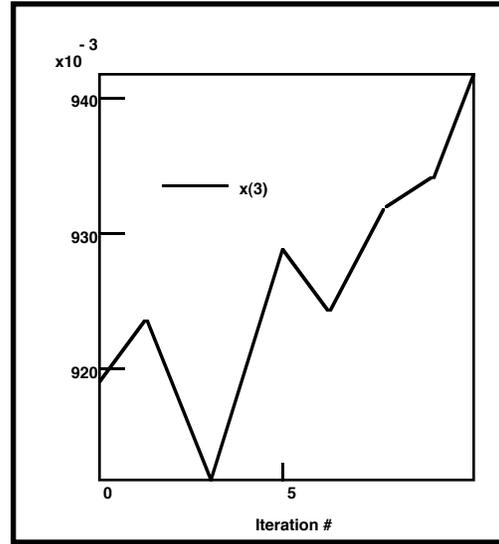


Figure 7. History of a design variable for the heart dipole problem.

IMPLEMENTATION WITH AN EXISTING FRAMEWORK

The third implementation (Fig. 8) merges the Web-based monitoring and control system with an existing framework called FIDO, the Framework for Interdisciplinary Design Optimization.¹ FIDO contains a fixed sequence of calls to the processes, as well as the iterations, in a single routine.

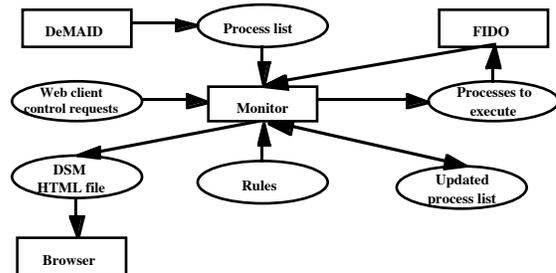


Figure 8. Web-based monitoring and control system with FIDO.

The initial plan to merge the two systems relied on the server program to call FIDO. This merger was not feasible to implement, so the routine in FIDO which calls the FIDO processes was modified to call a monitor program similar to

the one from the initial system. In addition, the process calls and associated code from FIDO were blocked as “if” statements which depended on the process being executed; and the code for iterations was removed. The file containing the “processes to execute” data contains the numbers of the processes ready for execution rather than the names.

The conceptual design of a High Speed Civil Transport (HSCT)¹ currently implemented in FIDO was selected to demonstrate this system. This design was later implemented in Hale’s Intelligent Multidisciplinary Aircraft Generation Environment (IMAGE) system.²¹ The HSCT problem is more complex than the heart dipole problem because it contains more processes and the processes are more complex in function. This implementation couples performance (Breguet range equation), structures (equivalent laminated plate), aero (doublet panel and wave drag), and propulsion (engine deck) analysis programs and their gradient calculations to an optimizer. The DSM shown in Figure 9 is identical in sequence to the current implementation of the HSCT problem in FIDO.

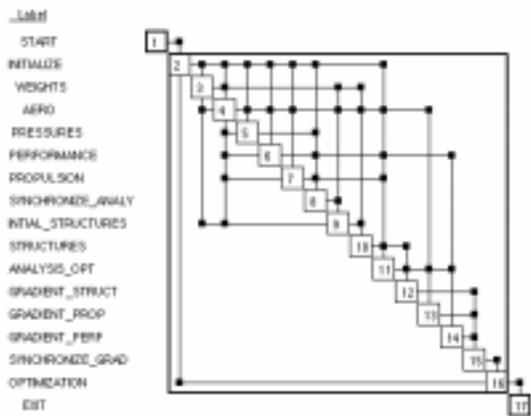


Figure 9. DSM for the HSCT problem.

There are two main iterations in this implementation. There is a weights iteration (processes 3-9) and an optimization iteration (processes 2-16). In the DSM, there are several feedbacks which could cause additional iterations. A control capability was added to allow the user to deactivate a feedback, thereby deleting an iteration

process. In this implementation, all feedbacks except 9 to 3 and 16 to 2 were deactivated so that the system could be tested against the current FIDO implementation. The current number of iterations is dependent on the coupling strength of the feedback, but can be overridden by the user.

This implementation has been tested through two optimization cycles of FIDO. Further testing needs to be done through a complete design cycle and results compared. In addition, the optimization code needs to be augmented to write the necessary files for interfacing with the design data display applets.

CONCLUDING REMARKS

This paper has described the evolution of a Web-based system for monitoring and controlling multidisciplinary design projects. This system uses a combination of World Wide Web technology and knowledge-based methods. A knowledge-based system is applied to determine the optimum process sequence. Given the sequence of processes and process coupling information for a particular design project, the knowledge base applies a set of problem-independent rules to determine the processes ready for execution. These processes are then executed by an independent framework.

A combination of Web programming techniques provides the means for displaying the status of design cycle processes as well as the optimization results, and for interfacing with Web pages to control the execution. Web technology provides a convenient and familiar interface for accessing process data and for interfacing with the design project.

In summary, several steps must be taken by a project team in order to use the Web-based system described in this paper. Integration with the knowledge-based system requires the following activities: (1) identify processes and coupling information of the design project, (2) apply DeMAID to optimize the process sequencing, and (3) obtain and possibly modify a framework to invoke the various processes identified as ready for

execution by the knowledge-based system. Integration with the Web tools requires the creation of process Web pages that contain links to information describing the processes. These pages also contain access applets to display design information. The system-created project Web page displays the project design structure matrix and links to the process Web pages.

The system has evolved to its present state by adding enhancements to enable the execution of increasingly more complex design problems. Because this system is still being tested and refined, it is not ready for distribution to the public. More testing is planned with the current system which will examine the effects of different process and iteration sequences on results and execution times. Final testing will be carried out with more complex models with the additional capability of executing processes in parallel when more than one process is ready for execution.

REFERENCES

- ¹Weston, R. P., Townsend, J. C., Eidson, T. M., and Gates, R. L., "A Distributed Computing Environment for Multidisciplinary Design," AIAA Paper 94-4372, Sept. 1994.
- ²Eldred, M. S., Outka, W. J., Bohnhoff, W. J., Witkowski, V. J., Ponslet, E. R., and Chen, K. S., "Optimization of Complex Mechanics Simulations with Object-Oriented Software Design," AIAA Paper 95-1433, April 1995.
- ³Hufford, G. S., Kingsley, G. M., and Harrand, V. J., "Requirements, Preliminary Design and Initial Implementation of a Generic Multi-Disciplinary Software Environment," AIAA Paper 97-0929, Jan. 1997.
- ⁴Blair, M., Bharatram, G., Canfield, R. A., "Designing a Blended Composite Wing and Fuselage," AIAA Paper 96-3995, Sept. 1996.
- ⁵Hale, M. A., Craig, J. I., Mistree, F., and Schrage, D. P., "DREAMS & IMAGE: A Model and Computer Implementation for Concurrent, Life-Cycle Design of Complex Systems," Concurrent Engineering: Research and Applications, Vol. 4, No. 2, June 1996, pp. 171-186.
- ⁶Multidisciplinary Integrated Design Assistant for Spacecrafts (MIDAS) Tutorial, Jet Propulsion Laboratory, Nov. 10, 1995.
- ⁷iSIGHT 2.0, Engineous Software Inc., 1996.
- ⁸Ridlon, S. A., "A Software Framework for Enabling Multidisciplinary Analysis and Optimization," AIAA Paper 96-4133, Sept. 1996.
- ⁹Daum, A., and Wolf, D. M., TRANSYS--A Multidisciplinary Software System for Preliminary Design, Analysis, and Evaluation of Space Transportation Systems. AIAA Paper No. 94-4342, Sept. 1994.
- ¹⁰Rahn, M., Schottle, U. M., and Messerschmid, E., Multidisciplinary Design Tool for System and Mission Optimization of Launch Vehicles. AIAA Paper No. 96-4130, Sept. 1996.
- ¹¹Computer Framework Workshop, MARC Building Auditorium, Georgia Institute of Technology, May 20-21, 1997. Web address: <http://web.cad.gatech.edu/image/cfw/>.
- ¹²Rogers, J. L., "DeMAID/GA-An Enhanced Design Manager's Aid for Intelligent Decomposition," AIAA Paper 96-4157, Sept. 1996.
- ¹³Rogers, J. L., McCulley, C. M., and Bloebaum, C. L., "Integrating a Genetic Algorithm into a Knowledge-Based System for Ordering Complex Design Processes," Proceedings of the Artificial Intelligence in Design '96 Conference, Kluwer Academic Publishers, Stanford, CA, June 1996, pp. 119-133.
- ¹⁴Steward, D. V., Systems Analysis and Management, Structure, Strategy and Design. Petrocelli Books, Inc., New York / Princeton, 1981.
- ¹⁵Savola, T., Westenbroek, A., Heck, J., Using HTML, Que Corporation, Indianapolis, IN, 1995.

¹⁶Giarratano J., "CLIPS User's Guide," NASA JSC - 25013, May 1993.

¹⁷Rogers, J. L., and Bloebaum, C. L., "Ordering Design Tasks Based on Coupling Strengths," AIAA Paper 94-4326, Sept. 1994.

¹⁸Padula, S. L., Alexandrov, N., and Green, L. L., "MDO Test Suite at NASA Langley Research Center," AIAA Paper 96-4028, Sept. 1996.

¹⁹Lemay, L., and Perkins, C. L., Teach Yourself Java™ in 21 Days, Sams.net Publishings, Indianapolis, IN, 1996.

²⁰Salas, A. O., and Rogers, J. L., "A Web-Based System for Monitoring and Controlling Multidisciplinary Design Projects," NASA TM - 97-206287, Dec. 1997.

²¹Hale, M. A., "An Open Computing Infrastructure that Facilitates Integrated Product and Process Development from a Decision-Based Perspective," Doctoral Dissertation, Georgia Institute of Technology, School of Aerospace Engineering, July, 1996.