NASA/TM-1998-207672

Integration of a CAD System Into an MDO Framework

J. C. Townsend
Langley Research Center, Hampton, Virginia

J. A. Samareh
Computer Sciences Corporation, Hampton, Virginia

R. P. Weston and W. E. Zorumski
Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

May 1998

Available from the following:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 487-4650

INTRODUCTION

Many design problems are multidisciplinary; that is, they require the coordination of information from a number of highly specialized disciplines. (For example, airplane design may include the disciplines of aerodynamics, structures, propulsion, controls, and manufacturing.) The point of view, design emphasis, and design approach of each discipline specialist can be quite different. Often, the practice has been for specialists to independently optimize each discipline with limited direct interaction or communication with others. Under the sponsorship of the federal High Performance Computing and Communications Program (HPCCP), the MultiDisciplinary Optimization Branch at the NASA Langley Research Center is investigating the use of a distributed heterogeneous computing system to facilitate communications, apply computer automation, and introduce parallel computing to produce a truly multidisciplinary design optimization (MDO) process. This concept is illustrated in figure 1.

As part of this work, NASA Langley has developed a heterogeneous distributed computing environment, called the Framework for Interdisciplinary Design Optimization (FIDO) (reference 1, 2, 3). The purpose of the FIDO project is to demonstrate the technical feasibility and the usefulness of its approach to optimizing the preliminary design of complex systems and also to provide a working environment for testing various optimization schemes. The FIDO system has now progressed beyond the feasibility stage and is being upgraded with major new capabilities. Because these capabilities require a more detailed geometry description than has been used so far, a commercial computer aided design (CAD) system will be used

This report first presents the philosophy behind some of the decisions that have shaped the FIDO system and then gives a brief case study of the problems encountered in integrating a CAD system into the FIDO system and proposed solutions. The report is an expanded version of a paper presented at the Engineering Foundation conference on Optimization in Industry, Palm Coast, FL, March 1997.

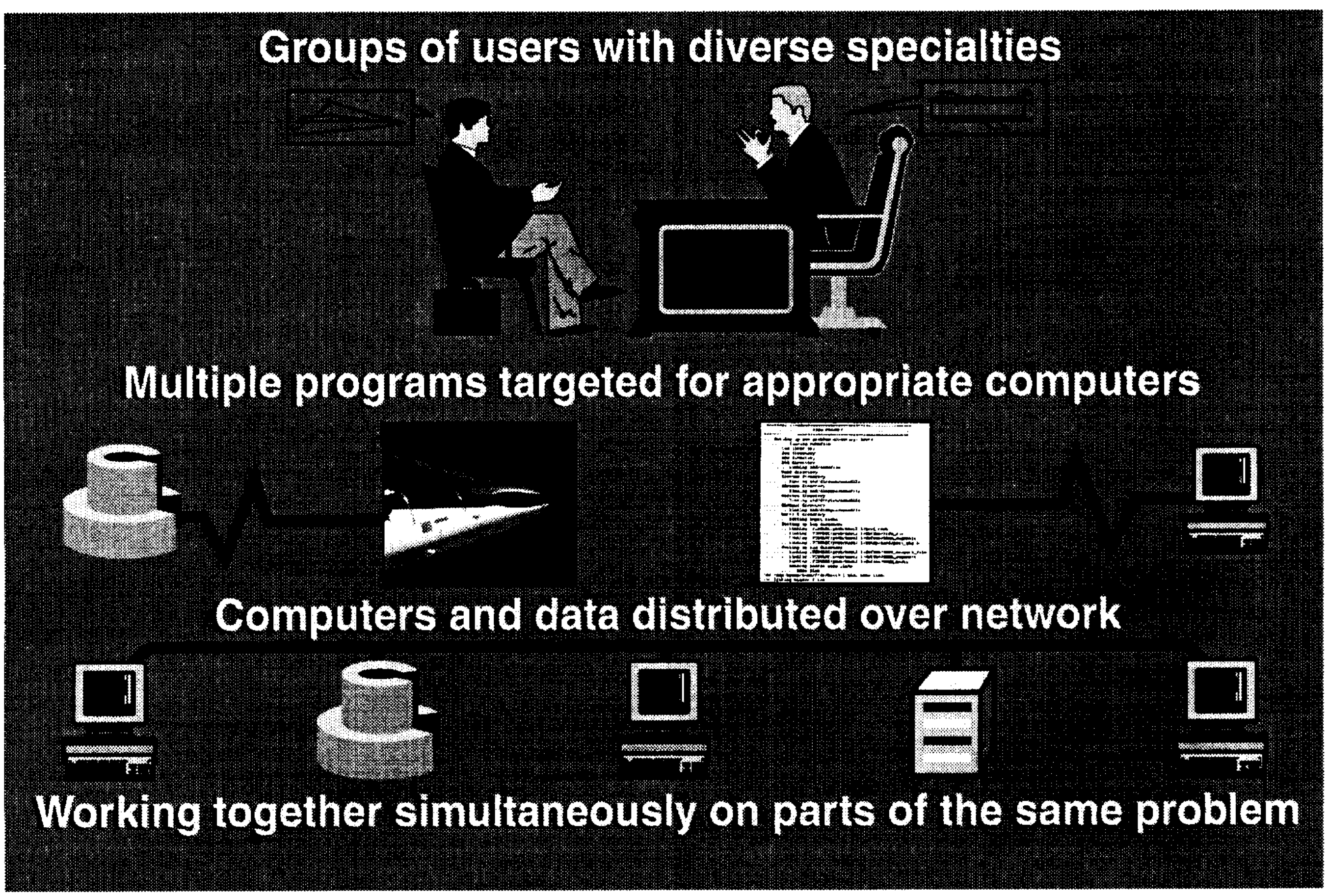

Figure 1. Concept of an MDO framework on HPCCP heterogeneous network of computers.

FIDO SYSTEM PHILOSOPHY

The FIDO system is a heterogeneous distributed computing environment being developed at the NASA Langley for optimizing complex designs that depend on several engineering disciplines for design analysis. The system has three purposes: demonstrate technical feasibility, demonstrate usefulness for selected applications, and provide a working environment for use by Langley researchers testing various optimization schemes. Philosophically, the FIDO system can be considered as a tool to be applied by a "design manager" who needs to improve a complex product design process. Basically, FIDO automates the coordination of analyses by the various disciplines (each on its assigned computer) into an overall optimization scheme, while allowing for visualization and steering of the process by the design manager.

Simplified test problem

For the purpose of developing the FIDO system, a very simplified test problem, based on preliminary design of a High Speed Civil Transport (HSCT, figure 2), was chosen. The disciplines included in the optimization are structures, aerodynamics, propulsion, performance, and an interdisciplinary interface (see system schematic, figure 3). The FIDO system was first demonstrated for a version of this design problem with fast, limited-fidelity discipline codes (equivalent plate structural analysis, linearized aerodynamic analysis, table lookup propulsion, and a range equation for performance fuel weight estimation), a geometry given by a set of points, a small number of design variables (on the order of ten), and a simple objective function. Recently it has been demonstrated with medium-fidelity structural (coarse-grain finite element analysis) and aerodynamic (supersonic marching Euler) codes.

Modularity

The FIDO system design is modular. The modules for such services as the graphical user interface (GUI), Executive control, Data Manager, Setup, and Spy (shown in figure 3) are intended to be independent of any particular application. In practice, some small changes may be necessary; for example, the introduction of higher fidelity discipline codes has required changes to system calls for transferring files to higher capacity computers. Obviously, the discipline modules and the code governing their use are much more dependent on the particular problem.

For integration into early FIDO versions, discipline codes were modified to make them modular. The legacy source codes, which were decades old and contained deeply embedded output and stop statements, were made to act like well-behaved library subroutine modules. For the latest FIDO version, "wrapper" technology has been

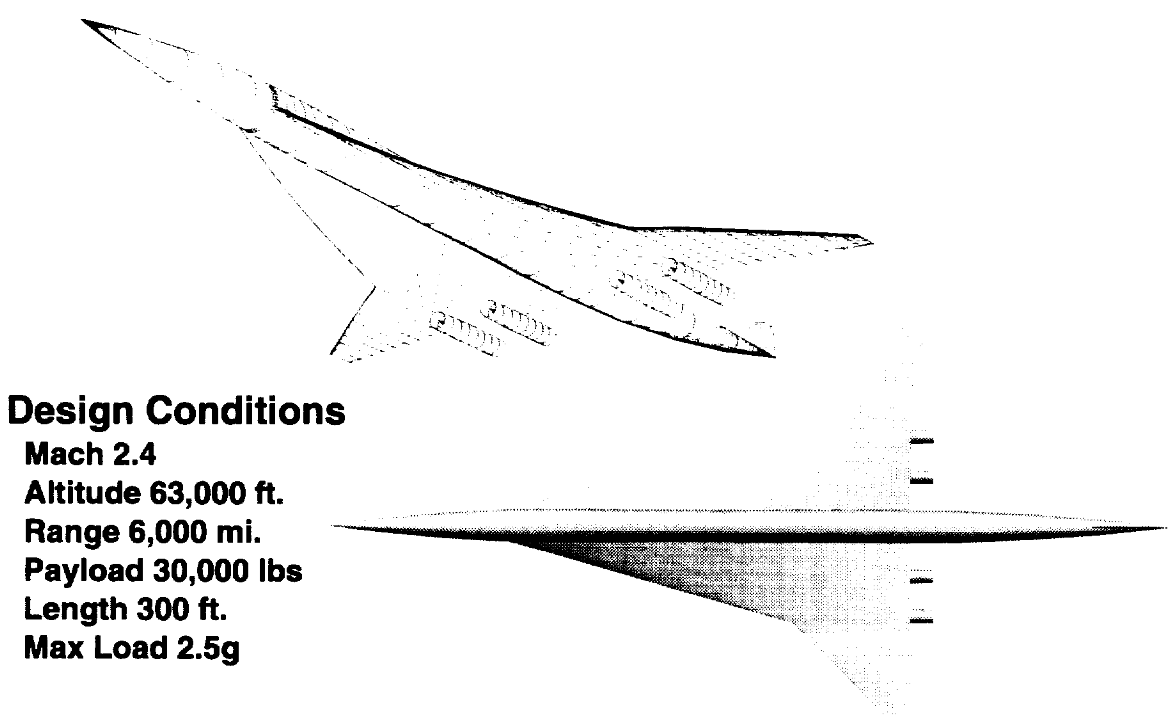

Figure 2. Sample FIDO application: High Speed Civil Transport (HSCT).

developed for unmodified legacy codes to properly format input, intercept unwanted output, and handle runtime errors that otherwise could cause unacceptable module behavior.

Communications

Each discipline analysis and auxiliary module is assigned, according to its computational needs and overall system balance, to an appropriate computer in a heterogeneous network of workstation, parallel, and super computers (see figure 1). Communication between the computers is handled by a Communications Library (reference 4, middle of figure 3) that is called from each module or, in the case of a legacy discipline code, its driver or wrapper.

The communication library contains functions designed to facilitate the communications within a general system of computer codes executed in a heterogeneous, distributed network of computers. This library allows the FIDO system to be programmed without recourse to the underlying message-passing primitives and minimizes the impact on FIDO of any changes in them. (Currently the primitives are the PVM system from Oak Ridge National Laboratory (reference 5).)

Coordination

Coordination of discipline analyses is provided by a problem-dependent Master module (figure 3). Currently, the code of this module must be rewritten for each specific application. For example, the simplified preliminary design of an HSCT (figure 2), the initial focus application of the FIDO project, requires code to perform a complex iterative looping behavior (figure 4). Ideally, the Master module would be rewritten to provide an interactive visual programming method for this linking of the modules containing the discipline codes. FIDO researchers are currently investigating the use of a commercial, general-purpose task automation system (iSIGHT[1] from Engineous Software Inc.) to provide a more readily modified coordination function.

Data base

All major data elements (individual items or file pointers) passed between modules go through and are stored in the central Data Manager (figure 3). Using the file pointers, data files are passed directly on request from the generating computer to the requesting computer. Because the communications library allows direct passing of data messages between the discipline comput-

[1]The use of this commercial product does not imply endorsement by NASA.

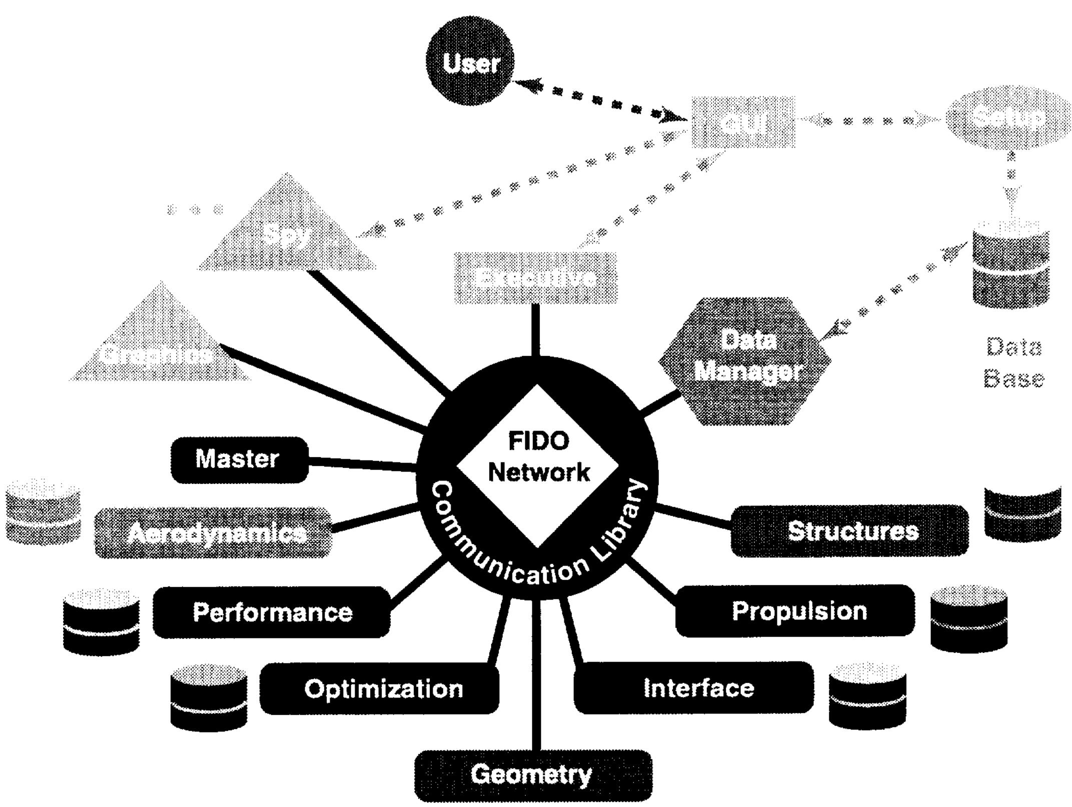

Figure 3. Schematic of the Framework for Interdisciplinary Design Optimization (FIDO) system.

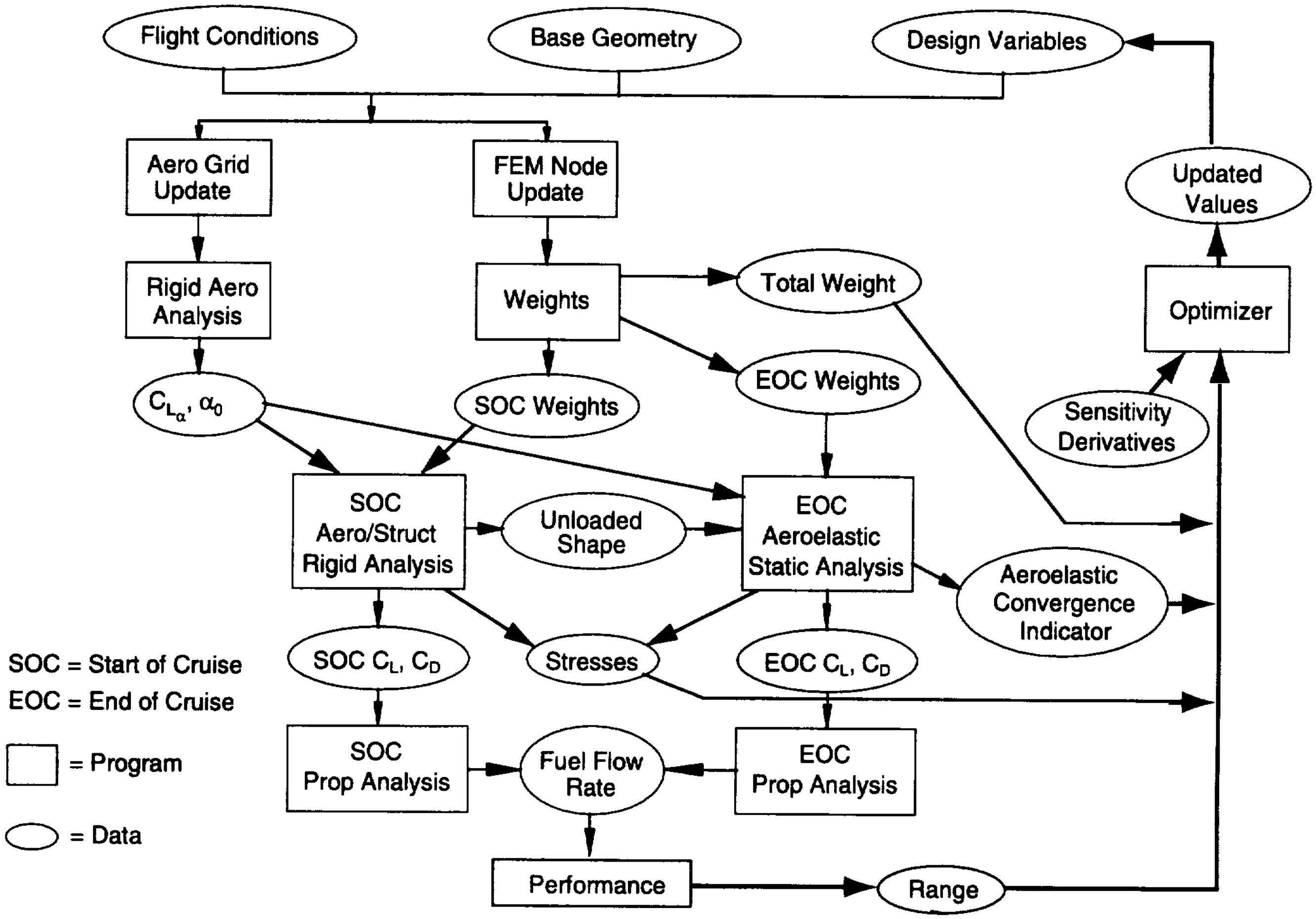

Figure 4. FIDO discipline interaction diagram for an HSCT application.

ers, direct communications of messages can be implemented if the increased efficiency warrants it.

User interface

The FIDO Graphical User Interface (GUI) gives the design manager control of the process and provides a way to view optimization progress and intermediate results. A feature of FIDO is that the viewing module (Spy in figure 3) can be invoked in multiple instances and from remote sites on the network in order to allow remote experts to evaluate the design graphically while it is in progress (figure 5).

Configuration files

Before execution of each case, the design manager typically invokes the GUI's Setup module to pick the system configuration and the initial conditions and constraints of the optimization process from a range of previously defined possibilities. These are contained in four configuration files that define the data in standardized formats: a file of the characteristics of all host machines to be considered; a file pairing the computational modules with the machines to be used for execution of a particular case; a file defining the base aircraft geometry, flight conditions, and design variable initial and bounding values; and a file specifying which segments are to be run in a debug or demonstration mode.

Upgrades

Current plans are to upgrade the FIDO system to use more realistic analysis modules, including both high-fidelity structure (adaptive refinement finite-element method) and aerodynamic (Navier-Stokes CFD) codes. A commercial computer aided design (CAD) system (Pro/ENGINEER® from Parametric Technology Corporation[2]) will be used to provide these codes with a unified, high-fidelity surface description. Because the purpose of the FIDO framework is to automate the MDO process, the CAD system must regenerate a new surface model each time the value of any geometric design variable is changed (as in figure 6).

[2]The use of this commercial product does not imply endorsement by NASA.

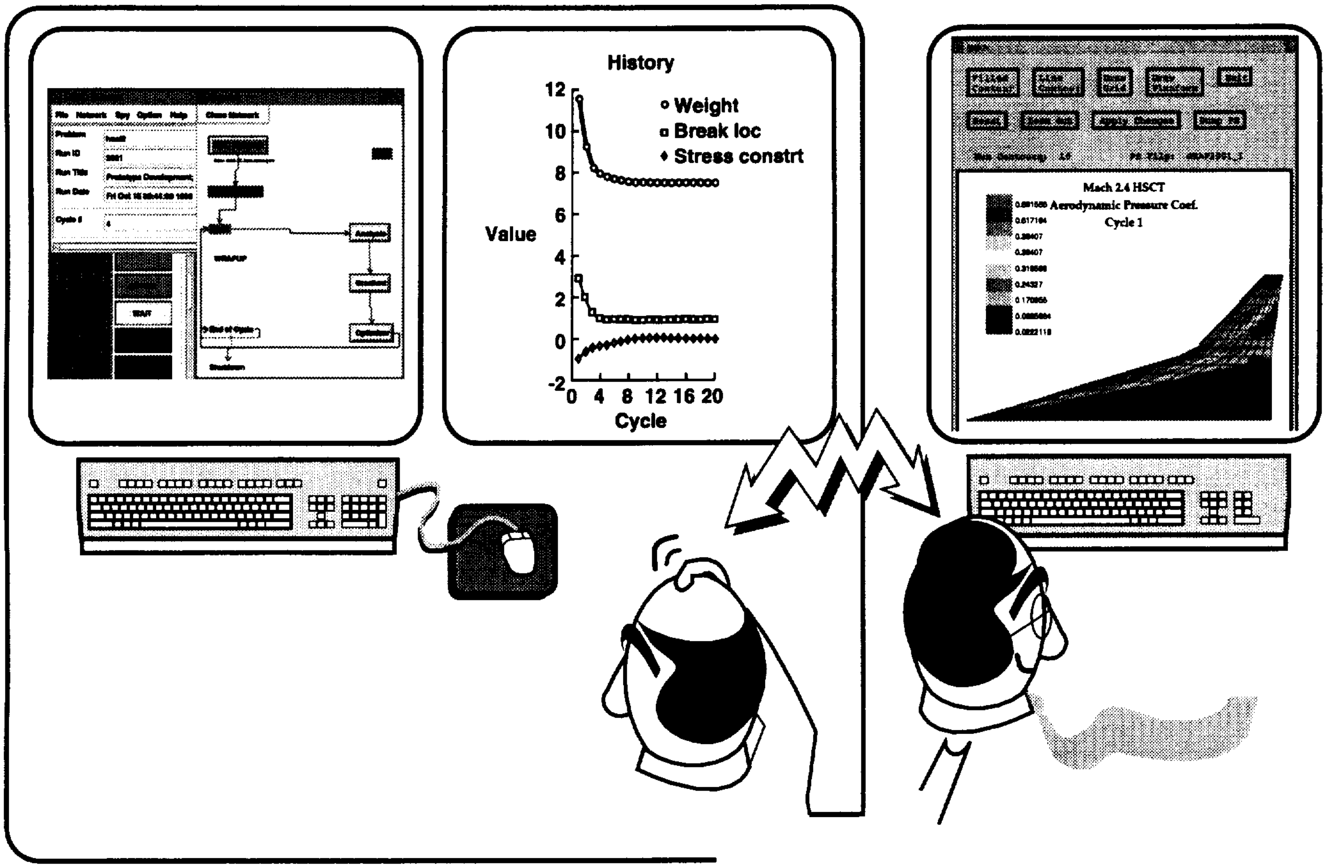

Figure 5. FIDO interface concept, showing design manager conference with discipline expert.

There are three main challenges to the integration of a CAD system into an environment for aerospace system optimization, such as FIDO: (1) devising a method for dealing with aeroelasticity, (2) allowing for replacement of the CAD system when desired, and (3) determining the sensitivity derivatives needed by the gradient-based optimizer. The approaches used to address these challenges are discussed in the following sections.

AEROELASTIC AND REPLACEMENT CHALLENGES

Aeroelasticity

In aircraft design, aeroelasticity involves the interaction of aerodynamic forces and structural deformations -- the airframe deformations depend on the loads imposed on the aircraft (especially on the wings) by the air flow, and the air flow is influenced by the deformations. Determining a deformed shape consistent with the corresponding aerodynamic loads is an iterative process.

Because the aeroelastic deformations do not change the parameters that define the *design* shape of the aircraft, it was decided to avoid the first challenge, modeling the deformations in the CAD system. Instead, the CAD system is used only for the outer design cycle, where it regenerates the aircraft model with updated values of its parametric design variables. The framework then copies the updated model to a separate, Non-Uniform Rational B-Spline (NURBS) database, and this NURBS model, which exactly matches the CAD model, is used for the inner aeroelastic-deformation and load-balance iterative loop in FIDO.

In this inner loop (figure 7), automated methods are used to obtain from the NURBS database (reference 6) the surface geometry information needed by the computational fluid dynamics (CFD) grid regeneration tool CSCMDO and finite element model (FEM) grid regeneration tool BSMART. The CFD analysis code CFL3D is used to compute aerodynamic forces that are converted into structural loads for computation of deformations by a computational structural mechanics (CSM) code, COMET-AR. These deformations are then converted to a NURBS representation and the NURBS geometry is modified to reflect the aeroelastically deformed model. This loop is repeated until the loads and deformed model

Figure 6. Parametric change of leading-edge sweep in Pro/ENGINEER® model of an HSCT.

shape converge, usually in three to five iterations. Then, the CFD and CSM results are applied to the overall optimization process, along with other discipline results, to determine a new set of values for the design variables (CAD parameters). These new values are used to regenerate the CAD model, and the whole process repeats until no further improvement in the objective is evident.

CAD replacement

It has become apparent that no one CAD system is uniquely applicable to MDO frameworks. Because most CAD systems can export the required NURBS geometry, the uncoupling of the CAD and aeroelastic databases as described above also eases the second challenge, because it simplifies the replacement of the CAD system when this is desired

SENSITIVITY DERIVATIVES CHALLENGE

The third major challenge to integration of the CAD system into FIDO is the determination of the sensitivity derivatives used by the gradient-based optimizer. Essentially, what is needed is the partial derivative of each computational grid point position with respect to each geometric design variable. These geometric derivatives can be chained together with derivatives in the discipline codes with respect to the grid points to obtain, eventually, sensitivities of constraints and objective functions with respect to the design variables. An example is the derivative of total aircraft weight with respect to wing leading-edge sweep angle, including contributions from the aerodynamic loads and the structural deformations. The optimizer then applies the overall derivatives to its algorithm for determining improved values of the design variables.

Derivatives using NURBS database

In some instances it will be possible to relate the NURBS control points to the design variables. For those design variables, the required derivatives can be computed without recourse to the CAD system. But, some alternate means will have to be found to compute the derivatives of design variables that cannot be so related.

Finite difference approximations

The usual direct approach is to apply an incremental change to each geometric design variable in turn and compute its finite difference approximations for the required derivatives. This method is being used by the

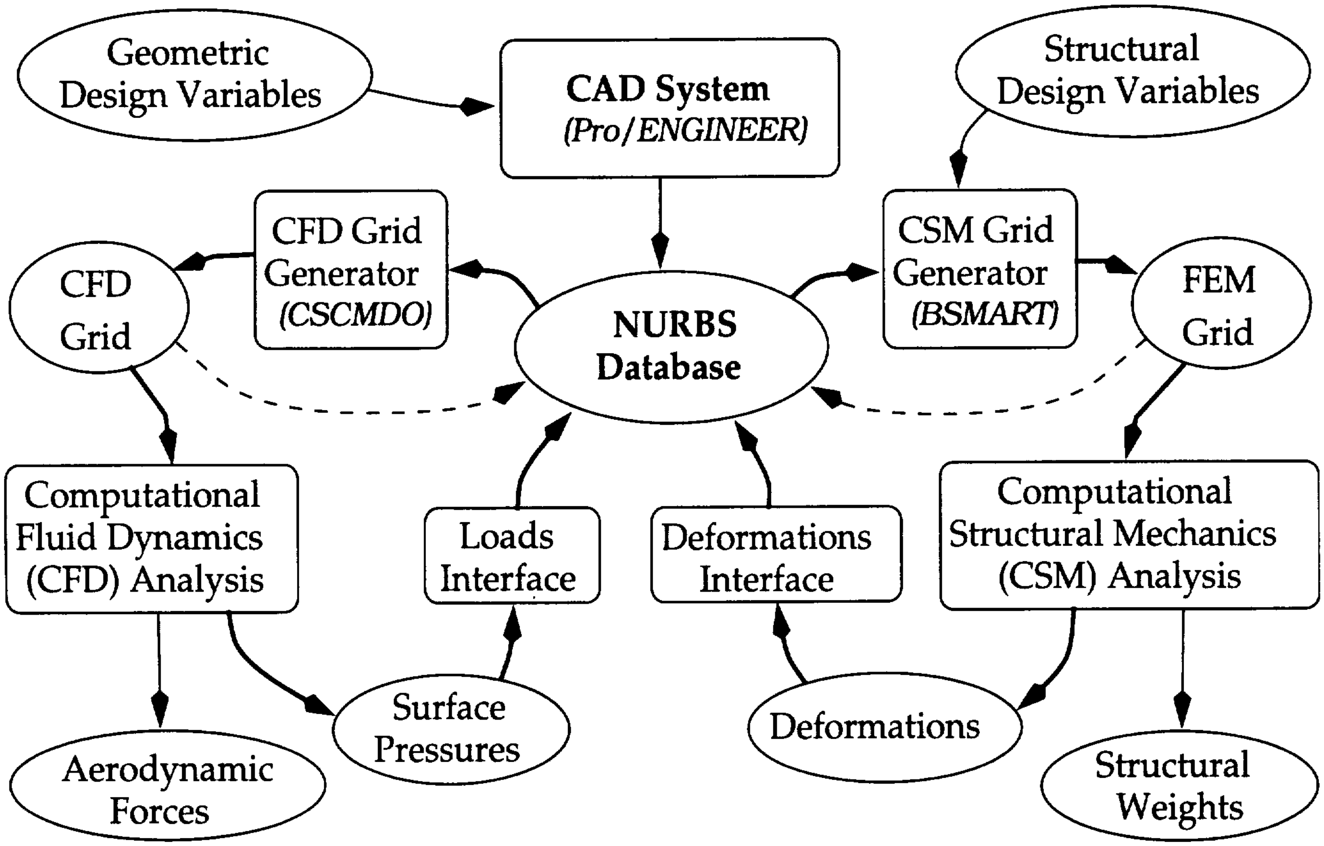

Figure 7. Aeroelastic iteration loop (heavy arrows) in the FIDO system.

FIDO project in the interim while pursuing analytic derivative methods.

The problem with finite differences is that the approximated value of a derivative sometimes is very sensitive to the size of the design variable increment used to compute it. The mathematical basis for the approximation is the Mean Value Theorem of calculus, which assumes that the function being approximated is continuous. This assumption is not strictly true for any computed function (due to round off and truncation errors) and may be far off (relatively speaking), depending on the nature of the geometry at a particular point. Even a relatively smooth surface geometry has regions of low and high sensitivity to any given design variable. Thus, the best size for the increment to use for each design variable depends on the derivative being computed; different increments may well be appropriate for different derivatives.

Obviously, trade-offs are involved in computing the finite difference approximations. If a single increment is chosen, the resulting poor approximation of derivatives can lead to poor performance of the optimizer. On the other hand, if many different increments are used for each design variable, the overall system performance may be poor because of the many additional regenerations required of the CAD system. These trade-offs and other issues relating to the use of finite difference approximations of geometric sensitivity derivatives are being studied at the Langley Research Center as the integration of Pro/ENGINEER with FIDO proceeds.

Analytic derivatives

Ideally, one would like to have analytically defined sensitivity derivatives for use in gradient-based optimization. Because surfaces in the CAD system are defined by a patchwork of analytic curves and surfaces, it is possible to produce analytic derivatives of those curves and surfaces. And, because the CAD system contains within its code the logic and equations that relate the surface patches to the design variables (CAD parameters), it is theoretically possible to produce analytic derivatives relative to the design variables.

A computer code, ADI-C (for Automatic DIfferentiation of C), which has been developed by researchers at Rice University and Argonne National Laboratory, takes a normal C code as its input and produces as output the same code supplemented with additional statements to compute the analytic derivatives of selected variables with respected to selected input variables. ADI-C (and its Fortran forerunner ADIFOR (reference 7)) have been tested and found to work reliably on such complex software as block-gridded CFD codes (reference 8)

Of course, the application of ADI-C requires access to the source code. Because the CAD system source code is proprietary, some kind of cooperative arrangement must be made with its owners in order for ADI-C to be applied. In preliminary talks, sales representatives have shown some interest relating to the application of ADI-C to the Pro/ENGINEER CAD system. It would appear to be a great advantage for a CAD system to be able to supply sensitivity derivatives to its users. However, it will ultimately take a business decision by a CAD system vendor to devote the necessary resources before this avenue to integrating a CAD system with MDO can be followed.

Transformation matrix formulation

There is another approach to obtaining analytic sensitivity derivatives that uses presently available information. The geometric sensitivity derivatives are essentially the linear transformation matrices used in producing the graphical displays of the geometry. Or, at least, the transformation matrices contain all the information needed to compute the derivatives, as in the case of rotations. For Pro/ENGINEER these transformation matrices can be accessed through a system library known as Pro/DEVELOP. Thus, the information needed is available without differentiating throughout the Pro/ENGINEER source code. At NASA Langley, we are currently looking into how to write the code necessary to pull out all the required matrices from the Pro/ENGINEER database and put them into sensitivity derivative form.

CONCLUDING REMARKS

More and more, the design of complex engineering products, such as aircraft, is coming to rely on computer-aided design systems. And, more and more, optimization methods are being applied to the design of complex engineering products in an effort to make them more efficient to manufacture, maintain, and use. Thus, there is a natural need to integrate computer-aided design systems with the emerging optimization environments, such as FIDO (the Framework for Interdisciplinary Design Optimization described herein). This report has discussed three issues arising in the integration and how they are being handled by the FIDO project. Of these, decoupling seems to handle the aeroelastic issue well and eases the CAD replacement, but the issue of deriving geometric sensitivity derivatives from the CAD system is more problematical. Which of the four proposed solutions (NURBS database, finite differences, CAD analytical derivatives, or transformation matrix analytical derivatives) turns out to be most effective in the long run remains to be seen.

REFERENCES

1. Townsend, J.C.; Weston, R.P., and Eidson, T.M.: A Programming Environment for Distributed Complex Computing. An Overview of the Framework for Interdisciplinary Design Optimization (FIDO) Project. NASA TM 109058, December 1993.
2. Weston, R.P.; Townsend, J.C.; Eidson, T.M.; and Gates, R.L.: A Distributed Computing Environment for Multidisciplinary Design. 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City, FL, 1994. (Published in AIAA CP9413, pp. 1091-1097; also available at NASA Langley Technical Report Server URL http://techreports.larc.nasa.gov/ltrs/1994.html)
3. Weston, Robert P.: FIDO - Framework for Interdisciplinary Design Optimization, A Programming Environment for Distributed Complex Computing. WWW URL http://hpccp-www.larc.nasa.gov/~fido/homepage.html, 1996.
4. Eidson, T.M.: A Programming Support Library for Distributed Memory Architectures. High Technology Report No. HTC-9501, High Technology Corporation, Hampton, VA, March 1995
5. Breshears, C.: PVM Guide. Joint Institute for Computational Science, University of Tennessee. (Available on the WWW at URL http://csep1.phy.ornl.gov/CSEP/PVM/PVM.html).
6. Samareh, Jamshid A.: Use OF CAD Geometry in MDO. 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellevue, WA, AIAA-96-3991, September, 1996, p. 12ff (Available at NASA Langley Technical Report Server URL http:// techreports.larc.nasa.gov/ltrs/1996-cit.html)
7. Bischof, Christian; Carle, Alan; Khademi, Peyvand; and Mauer, Andrew: The ADIFOR 2.0 System for the Automatic Differentiation of Fortran 77 Programs. CRPC-TR94491 or ANL/MCS-P481-1194, 1994. (See also WWW URL http://www.cs.rice.edu/~adifor/).
8. Taylor III, A.C., Oloso, A., and Newman III, J.C., "CFL3D.ADII (Version 2.0): An Efficient, Accurate, General-Purpose Code for Flow Shape-Sensitivity Analysis," Proc., 15th AIAA Applied Aero. Conf., AIAA Paper 97-2204, June 1997.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 07704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	May 1998	Technical Memorandum

4. TITLE AND SUBTITLE
Integration of a CAD System Into an MDO Framework

5. FUNDING NUMBERS
509-10-11-01

6. AUTHOR(S)
Townsend, J. C., Samareh, J. A., Weston, R. P., and Zorumski, W. E.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER
L-17726

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA/TM-1998-207672

11. SUPPLEMENTARY NOTES
Townsend, Weston, and Zorumski: Langley Research Center, Hampton, VA; Samareh: Computer Sciences Corporation, Hampton, VA. Based on presentation to Engineering Foundation conference on Optimization in Industry, March 23-27, 1997, in Palm Coast, FL

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified–Unlimited
Subject Category 02 Distribution: Nonstandard
Availability: NASA CASI (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
NASA Langley has developed a heterogeneous distributed computing environment, called the Framework for Interdisciplinary Design Optimization, or FIDO. Its purpose has been to demonstrate framework technical feasibility and usefulness for optimizing the preliminary design of complex systems and to provide a working environment for testing optimization schemes. Its initial implementation has been for a simplified model of preliminary design of a high-speed civil transport. Upgrades being considered for the FIDO system include a more complete geometry description, required by high-fidelity aerodynamics and structures codes and based on a commercial Computer Aided Design (CAD) system. This report presents the philosophy behind some of the decisions that have shaped the FIDO system and gives a brief case study of the problems and successes encountered in integrating a CAD system into the FIDO framework.

14. SUBJECT TERMS
multidisciplinary optimization frameworks
computer-aided design

15. NUMBER OF PAGES
13

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102