10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference
30 August - 1 September 2004, Albany, New York

AIAA 2004-4567

MDOPT – A Multidisciplinary Design Optimization System Using Higher Order Analysis Codes

Stephen T. LeDoux*, William W. Herling†, Gasper J. Fatta‡
The Boeing Company, Seattle, Washington, 98124-2207

and

Robert R. Ratcliff§
FutureTek Net Services, Austin, Texas

A Multidisciplinary Design Optimization (MDOPT) framework has been developed for air vehicle design and analysis. The developed MDOPT system contains a collection of technology modules for performing optimization studies by means of a Graphical User Interface (GUI), and combining robust numerical optimization schemes with higher order computational analysis. A variety of multidisciplinary objective and constraint functions are available, including aerodynamic, weight, mission performance, and stability and control characteristics. This paper is intended to provide an overview of the MDOPT development and test case results for a generic fighter configuration, funded under an Air Force Contract, F33615-98-2-3014, with Boeing cost match funds. The period of development was established with contract activity beginning in September 1998 and ending April 2002. Continued MDOPT development and design applications are being pursued within Boeing using internal funding.

I. Introduction

The overriding objective in creation of the MDOPT system was to provide a MDO framework whereby significant reductions in design cycle times and costs can be demonstrated, along with significant improvements in design quality for complex air vehicle configurations. The aim of development was to produce a solution that would gain acceptance and provide real-world usefulness, supporting tradeoff analyses to balance the demands of product performance and product cost as an objective function, without compromising flight safety. Contributing to the realization of these primary objectives were several system design goals that define the operational characteristics of MDOPT. A modular and open computing architecture was, in particular, crucial for the acceptance of MDOPT by a heterogeneous community of users. This design goal promotes the attainment of several key system characteristics: (1) scalability, to enable the user to frame application of the system to fit available schedule requirements and computing resource availabilities for the optimization problem; (2) flexibility, to enable the user to choose from a variety of solvers and other computer aided engineering tools; and (3) extensibility, to enable the system to grow through refinement of existing capabilities and the incorporation of new ones (e.g., the incremental incorporation of additional analysis disciplines or enhanced geometric complexity.)

In keeping with these design goals MDOPT can be described from the perspective of two view points, the system architecture described in Figure 1 and the system process described in Figure 2 below. The system is comprised of six main modules as shown, where the executive or MDO Manager (MDOM) controls the overall processes and provides common utilities necessary for general system functionality. The general components of the MDOM are illustrated. The primary component of the MDOM with which

* Senior Specialist Engineer, Member AIAA
† Associate Technical Fellow, Senior Member AIAA
‡ Manager Aerodynamics CFD, Senior Member AIAA
§ Senior Specialist Engineer, Member AIAA

Copyright © 2004 by The Boeing Company. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

the user will interact is the Graphical User Interface (GUI). Technology discipline domains provide analysis services of candidate configuration geometries. Current MDOPT implementation provides support for three technology domains, aerodynamics, structures and stability and control (S&C). The Inter-domain Communication Facility (ICF) maintains the underlying process control. It is constructed using TCL[1] scripts, the MICO[2] CORBA Orb and the Combat[3] TCL to CORBA[4] Bridge. Transfer of user input information is achieved through a master namelist file and parser utilities that extract or modify namelist parameter definitions as needed throughout the process. This namelist file also acts as the conduit between the GUI, the ICF, the Database Management Facility (DMF) and the discipline domain control scripts. All persistent user data, i.e. data created by the system, is stored into a MDOPT database via DMF utilities and MYSQL[5], with the exception of large binary computational model files (e.g. flow solver restart files), which are maintained separately within the system directory structure. During interactive execution, the user inputs the required data via the GUI, which edits the namelist file accordingly and directs the conduct of the optimization through the ICF. At several points within the optimization process the user has access to the database and may input data manually for given discipline data or for a particular case as represented by the man in the loop figures.

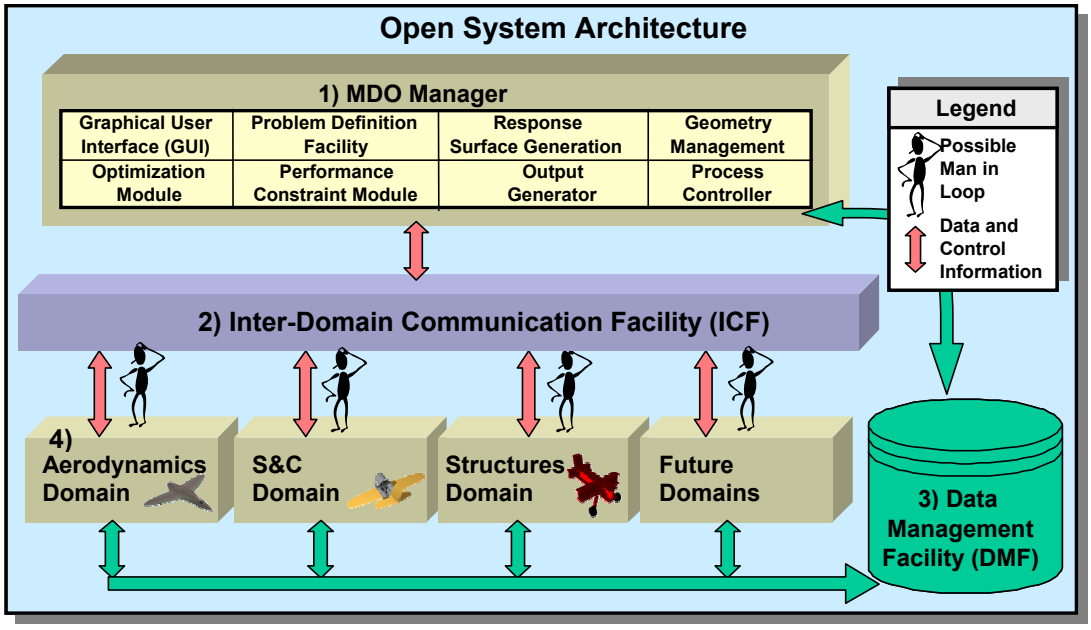

Figure 1 Diagram of the System Architecture.

The MDOPT system was developed as an enhancement and extension to the original 3DOPT[6] system. The main components of the 3DOPT system and process steps in an optimization are shown in Figure 2. This illustration provides a general outline of the steps required in performing an optimization with MDOPT. Following a clockwise direction, first the geometry is input in to the system and surface grids or lofts are created for the input geometry. Next the user defines the design variables and selects a design of experiments (DOE) experiment. For each design point in the experiment geometry perturbations are created and run through each of the discipline analysis codes. Geometric constraint checks are performed and stored into the database. Next Interpolated response surfaces (IRS) are created for the constraints and objective functions. Optimization is then performed on these IRS models and the final optimum geometry and design vector is output.

While MDOPT represents a major revision and extension to the system architecture used within 3DOPT, the basic 3DOPT system processes are the same and provided the template for additional processes required for new analysis capability. The components of the 3DOPT system that now make up a significant portion of the MDOPT MDO Manager (MDOM) as outlined in green in Figure 2. The field gridding and flow solver components blocked out in blue are now part of the aerodynamics domain within

MDOPT. This block can be thought of as a general discipline analysis block where structures and S&C analyses also take place as part of the optimization process. Once initial user inputs have been completed the system can proceed in an automated fashion. Parallelization of the optimization process has been implemented within most domains providing a capability to utilize large scale, multiple CPU computing devices in a computationally efficient manner.

The initial MDOPT system release provides a 3-D vehicle shape (outer mold line, OML) design optimization capability for wing alone, wing in the presence of a body, or wing with body and a single longitudinal controller (canard or conventional horizontal tail). Global or local direct driven design optimizations may be completed using a variety of multidisciplinary objective and constraint functions, including mission performance metrics, aerodynamic characteristics, weight trends and stability and control characteristics.

Remaining sections of this paper will provide details of the six components outlined, with a review of the system functional test case, designed to demonstrate the systems initial capabilities.

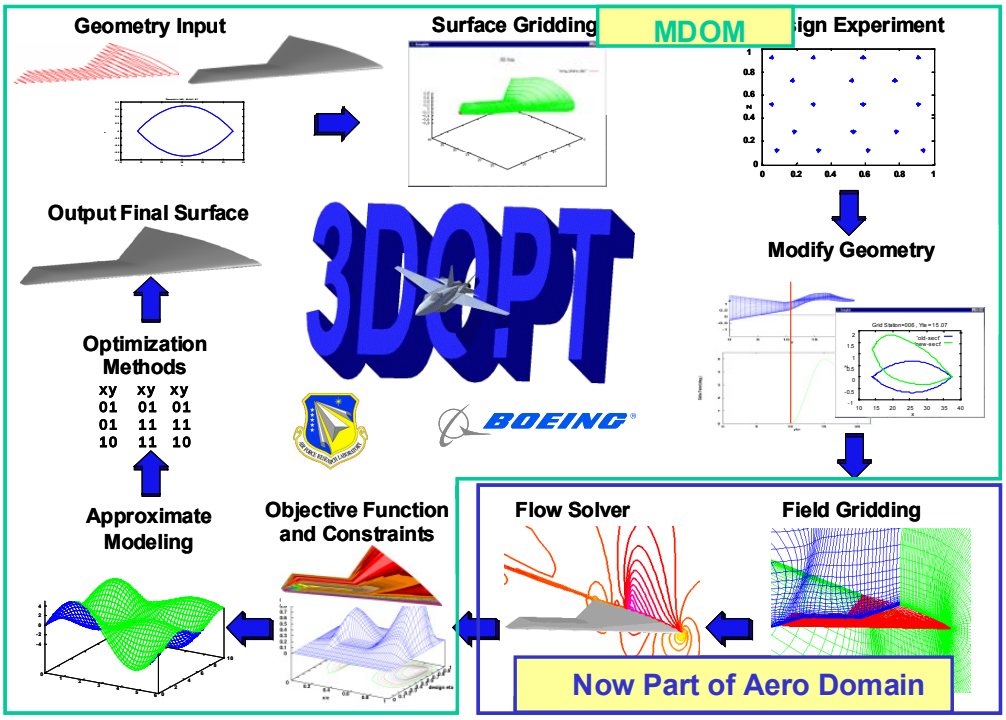

Figure 2 3DOPT process flow as applied to MDOPT system.

II. Inter-domain Communication Facility (ICF)

The inter-domain communication facility (ICF) shown in Figure 3 is an enterprise-level framework that supports loosely and tightly coupled distributed network access to engineering information and analysis services. The ICF also supports arbitrary user-defined workflows and complex binary data structures. It interoperates well with processes initiated on SMP and clustered computing environments via batch queuing systems such as Portable Batch System (PBS[7]).

Given that in a large distributed engineering environment each discipline maintains specialized domain knowledge and might prefer control over their processes, one of the design goals of the ICF was to allow access to these services without forcing each organization to relinquish control over their processes. The ICF is built upon the foundation of the flexible, widely adopted, and stable CORBA[4] standard developed over the last decade. Presumably by using a standard like CORBA engineering and finance departments providing information and analysis services would be more likely to make the investment in exposing their services without the fear of vendor-lock-in or continuous retooling caused by being early adopters of various fledgling communication frameworks. During the development of the CORBA standard in the 90's,

CORBA implementations had a history of being expensive, non-interoperable, and overly complicated. With the stabilization of the standard in the late 90's and the current availability of many free implementations[8] with bindings for C, C++, Java, Tcl, Python, Perl, and powerful developer-friendly scriptable solutions, these growing pains have practically disappeared. The CORBA implementations chosen for the current ICF are MICO[2] and Combat[3].

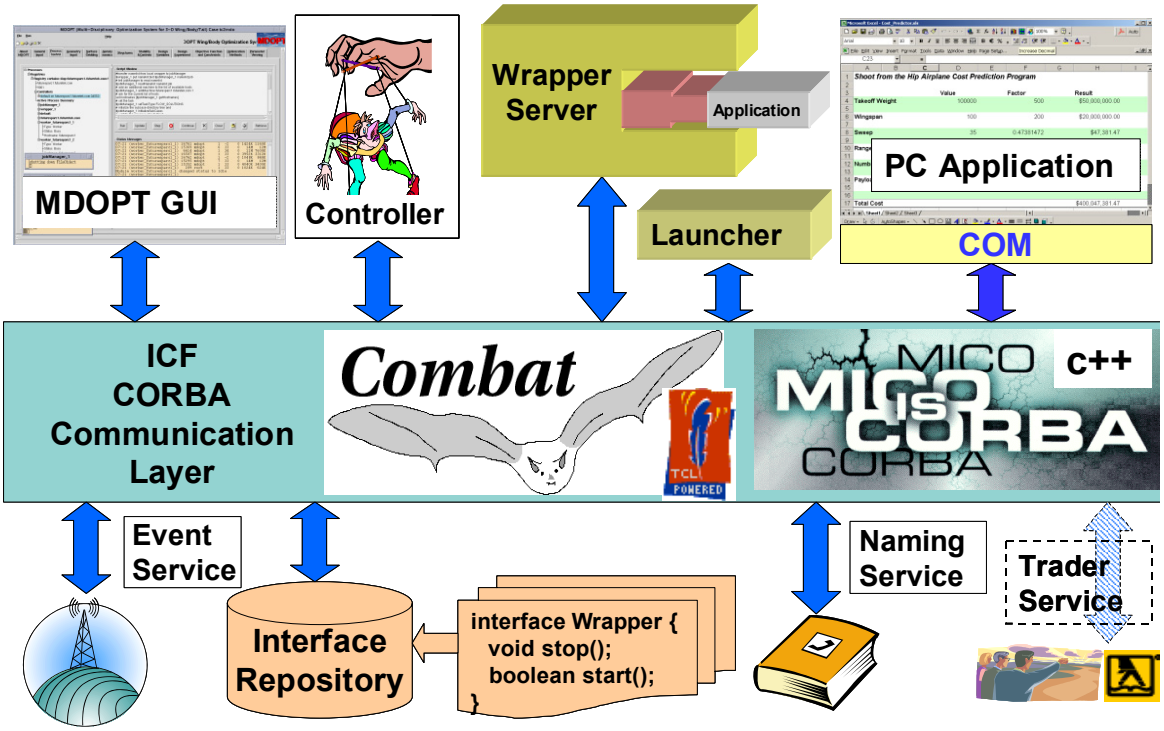

Figure 3 Architectural Overview of the ICF

Referring to Figure 3, the ICF leverages three of the standard CORBA services bundled with the MICO CORBA distribution. The first service is the COSS Naming Service that is used as a "white pages" style lookup of the currently running processes. The processes are organized hierarchically and can be keyed off a given userid. The second service is the interface repository that houses the functional and data type definitions of the available ICF services. The GUI dynamically queries the interface repository for these definitions and provides them graphically to the user to allow him to define a workflow in the script editor. The ICF clients and servers built using Combat also retrieve the required interface and data type definitions at run-time from the interface repository. The third standard CORBA service used is the Event Service[9]. This service provides an asynchronous publish and subscribe messaging capability similar in behavior to a radio station broadcast. Clients (providers) can push information to a given channel on the event server and numerous interested consumers will receive the information. The ICF leverages this capability to allow longer running processes to broadcast their status and analysis output information to multiple GUI clients. A given GUI can tap into the information stream at any point in time to view the current state of the MDO processes. The system was designed such that no state information was stored within the GUI thus allowing the GUI to be shutdown and restarted at will without any impact on process execution; the GUI merely provides a portal to the executing process flow. Multiple GUIs can also be connected simultaneously from different computers.

Given that the ICF processes are autonomous CORBA compliant servers, arbitrary process flows can be choreographed between them by writing client code using any of the available CORBA language bindings. For MDOPT, an innovative open-source binding for the Tcl language called Combat, written by a developer in Germany named Frank Pilhofer, was utilized to allow the user to script process flows from within the MDOPT GUI as shown in Figure 4. By leveraging a mature full-featured language like Tcl,

complex process flows are flexibly supported in a more natural way than perhaps a custom language solution might allow.

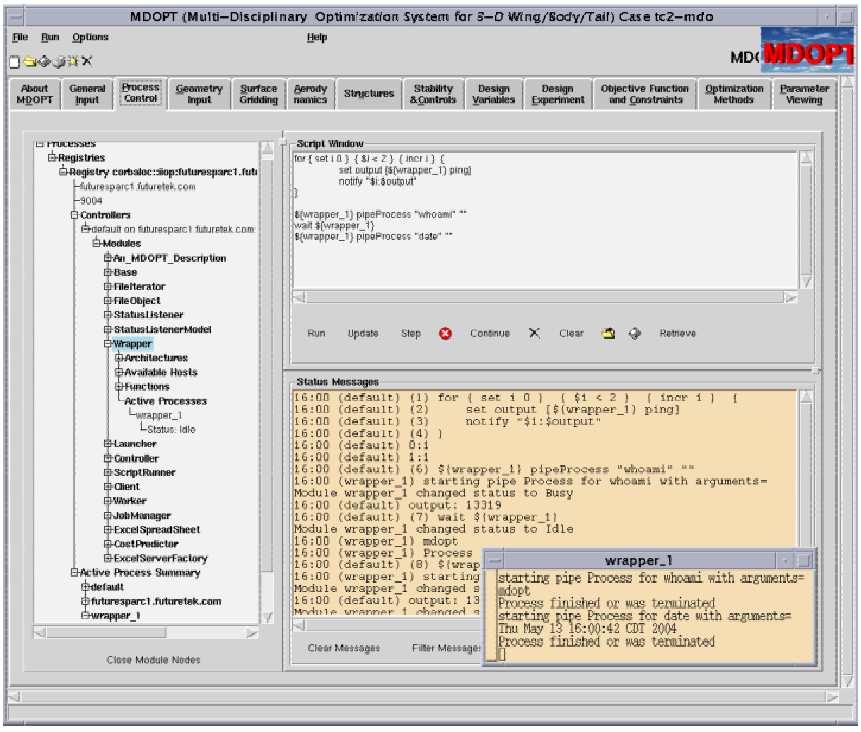

Figure 4 An example of defining a simple process flow in the ICF GUI

Combat was also used to create various ICF server processes. For instance, a service called the "Wrapper" was developed to allow loosely coupled integration of existing engineering applications. As depicted in Figure 5, this service launches the desired application via a Unix pipe. The Wrapper allows remote control over the life-cycle of the launched application, redirects the standard output from the launched application to the event server for broadcasting to the GUI, provides file transfer capabilities to any other Wrapper and status information to be queried and broadcast.

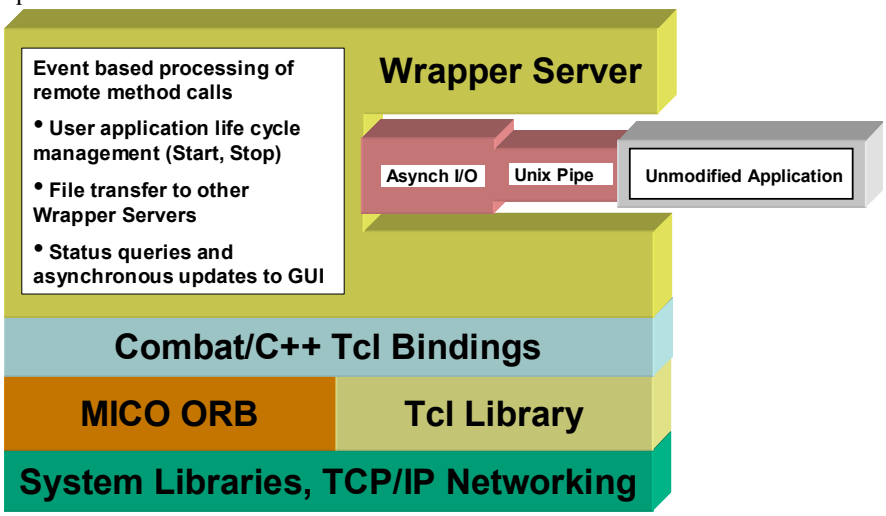

Figure 5 Wrapper Server

Other ICF services developed with Combat are the Controller, Launcher, and Job Management system. The Controller provides a convenient façade interface that exposes various factory methods for generating ICF Script Runner, Launcher, Wrapper and other services.

The Script Runners are used by the GUI to execute and step through the ICF scripts generated by the user. The GUI interacts with the Script Runner via the ICF. The Script Runners reside in their own process threads within the Controller process to avoid blocking the entire server during a long running ICF script. One of the benefits of running the scripts in a server process rather than within the GUI is that the GUI can be disconnected as soon as the script is initiated.

The Launcher Service is a simple server that starts registered ICF services only on the machine that it is running. The Launcher Service is started on the remote machine by the Controller via standard remote shell or manually by the user on the command line. The major benefit of this service is that remote shell does not need to be available on all the machines accessible to the ICF such as Windows computers or those with remote shell disabled.

The MDO process requires a large number of computer aided engineering solutions, most of which can be run in parallel. In the past, the user was responsible for doling out the solutions to the various computers available to him. It was difficult to know how to best distribute the solutions for the most efficient turnaround and to manage the associated bookkeeping. A distributed job management system was developed to alleviate this problem using the facilities of the ICF. This capability allows the user to start a Job Manager process from the GUI that efficiently parcels out jobs from a queue that it manages to workers started by the Job Manager or manually on selected machines. Once initiated, the workers request a task from the Job Manager. The workers continue to request jobs until the queue is empty after which they are terminated. This approach allows efficient use of heterogeneous computing resources including computers with managed batch queuing systems such as PBS and Linux clusters.

Microsoft Excel spreadsheets are significant sources of information from disciplines such as finance or airplane weights. These spreadsheets can be accessed through the ICF in a loosely coupled or tightly coupled fashion. (A Tcl extension called tcom[10] is used to communicate programmatically with Microsoft Windows programs such as Excel via COM.) In the first case, an instance of a generic ExcelSpreadSheet service that instantiates the specified spreadsheet is created through the ExcelServerFactory service that is running on the PC where the spreadsheets are located. The spreadsheet's cell data are accessed by row and column addresses programmatically from within an ICF script. This approach has the advantage that spreadsheets can be quickly integrated, but it can lead to a brittle integration since the ICF scripts have to change any time a given field is moved to another cell location or it's name changes.. Another supported approach that provides a tighter integration and more intuitive interface with the spreadsheet is to create an extension of the ExcelSpreadSheet service through inheritance that exposes the information via specific "getter" and "setter" methods with names patterned off the desired fields or aliases. This allows the new service to shelter the developers of ICF scripts from the structural changes in the spreadsheet. Furthermore, the new service and the more intuitive methods will appear in the GUI. An example of interacting with spreadsheet using this latter approach is shown in Figure 6.

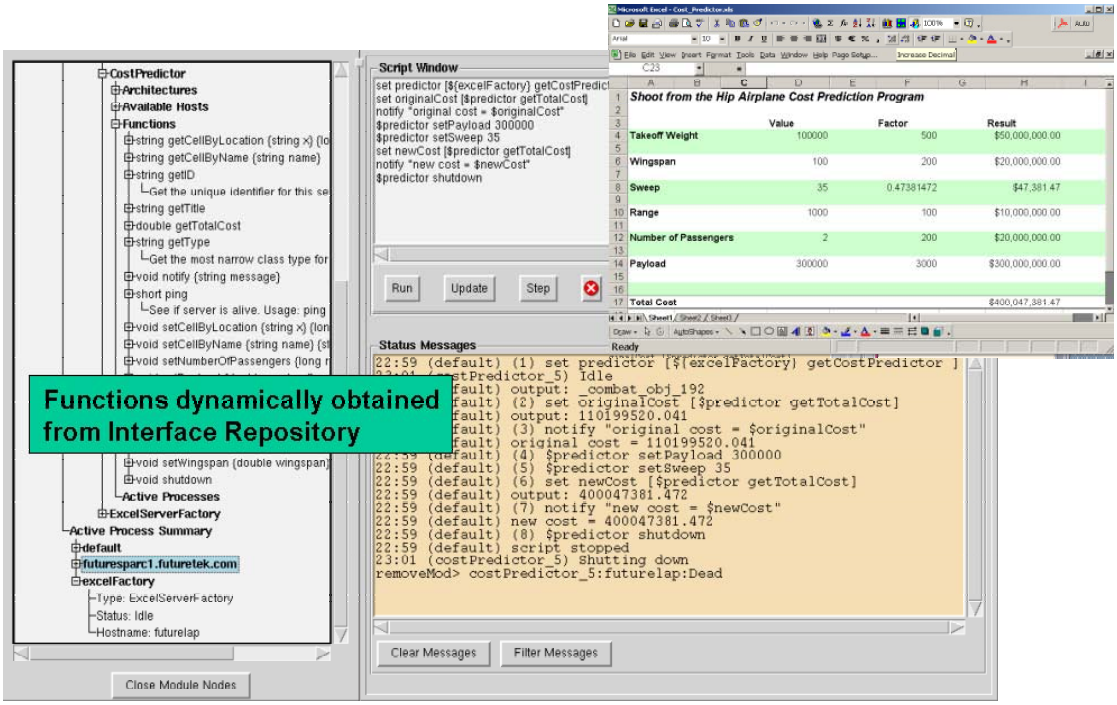

Figure 6 An example of interacting with an MS-Excel Spreadsheet

III. The Data Management Facility (DMF)

The DMF is a framework that allows networked access to a centralized relational multi-user database. The current DMF is built upon the free open source relational database MySQL and a Tcl extension called MySQLTcl[11]. MySQL is a fast, networked, multi-user database that supports most of the SQL 92 standard. MySQLTcl provides the Tcl bindings to the MySQL communication library.

The DMF architecture allows data from non-automated and automated engineering processes to be included in the optimization process from any where on the network. Currently, the DMF is used to store all of the scalar and vector data needed by the different MDOPT processes. For instance, objective and constraint values required by the response surface generation routines are calculated from the raw data stored in the database. Data can be stored and edited either interactively through the DMF GUI as shown in Figure 7 or via the Unix-style command line interface that can be embedded into shell scripts. The command line interface allows data stored locally in a file with a FORTRAN-style namelist format to be stored in the database and the data stored in the database to be retrieved locally to a namelist file. The DMF functionality can also be directly accessed programmatically from within Tcl scripts via the object-oriented DMF library.

The DMF relies on a relational database. A typical relational database organizes data by columns in various interconnected tables. The entity relational diagram, ERD, shown in Figure 8 depicts the tables and their relationships to each other for the DMF database. All the desired computed data for the solutions are stored in one table called VarValues. This type of table structure was designed to allow the database to scale to any number of variables associated with data vectors of varying lengths without having to create new table columns for each variable. Since MDOPT allows the user to create variables on the fly, it was important that the database schema support new variables without requiring table structure changes. The design tradeoff is that SQL queries are a little more complicated and performance can be slower, but through the careful use of table indexes, data retrieval from the DMF is fast even with such a simple table structure.

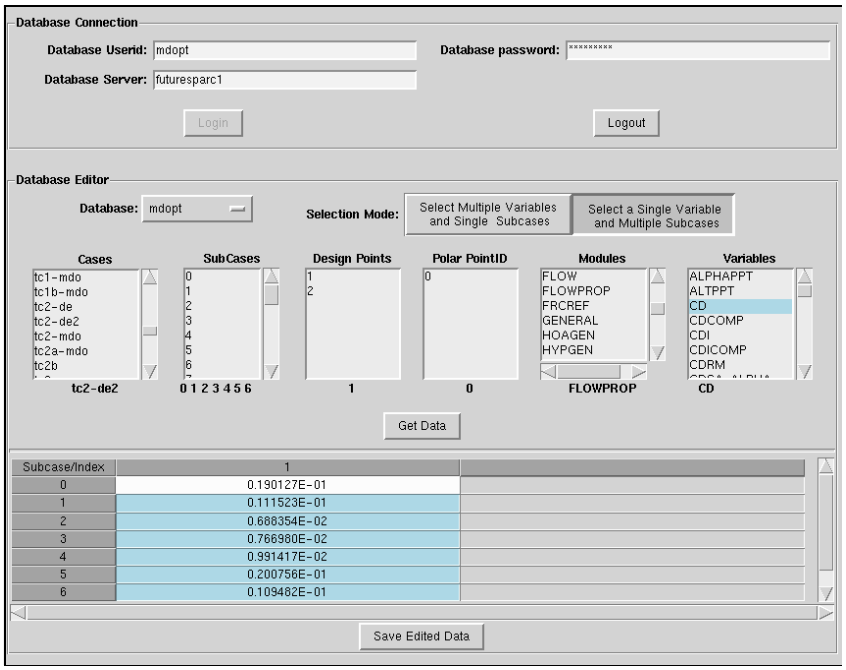

Figure 7 DMF GUI

Most of the other tables provide metadata associated with the computed data values. For example, the Variables table contains descriptions of each variable including its name, type, units, allowable range, and a human readable description. This information is available from the DMF GUI. The Cases table's columns provide a way to ID a particular solution. The current scheme reflects the domain-specific organizational structure of the data. The Status table provides a simple way to track the status for each step of the solution process for a given configuration associated with a given PointID.

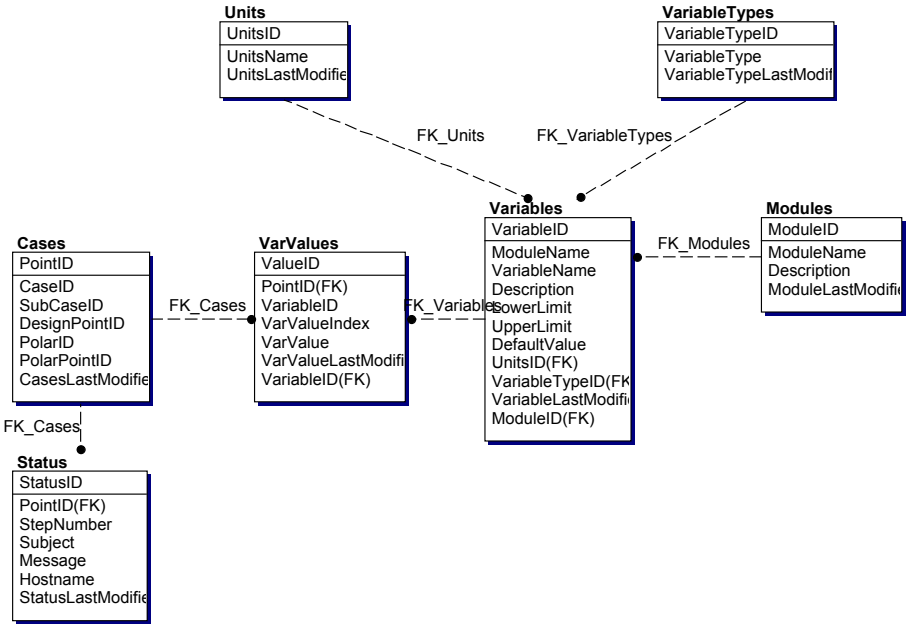

Figure 8 Database Schema for DMF

IV. Discipline Domains

As part of the initial implementation of the MDOPT framework three discipline domains were included in system development. These included aerodynamics, structures, and stability and controls. In a perfect system one would include all disciplines impacting the final performance of the configuration, however due to budget limitations and schedule constraints only these three domains were included in the first release. MDOPT does however include all the necessary utilities for extension to other domains.

Aerodynamics

The MDOPT aerodynamics domain provides five aerodynamic functions: field gridding, flow solution, aero-structural interaction, aero database generation, and performance analysis. The primary function of these utilities within the MDOPT system is to provide the system database with aerodynamic data: configuration and component force and moment coefficients, stability derivatives and mission performance metrics. The extent of data required will depend on the type of optimization problem to be solved. For example, if an aerodynamic optimization is to be performed, e.g., minimize drag then performance inputs and aero database inputs are not required. However, if a mission performance based optimization is to be performed, e.g., maximize range then inputs for all segments will be required. Two system grid topologies are supported for flow analysis within MDOPT. A single-block topology for wing alone or wing-body configurations can be used with TLNS3D or OVERFLOW. A multi-block overset topology, as shown in Figure 9 for wing-body-tail or wing-body configurations can be used with OVERFLOW. For additional topologies the system is extensible enough to allow user's gridding and flow processes to be integrated.

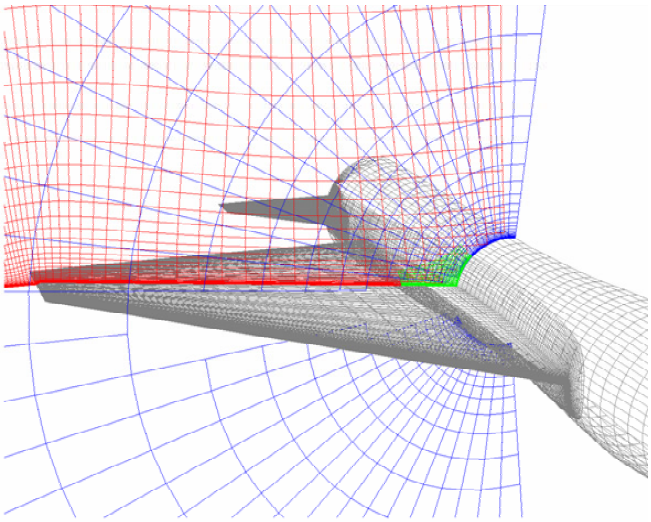

Figure 9 Example MDOPT overset grid topology.

The MDOPT system provides four flow analysis methods: A502 (public domain version PanAir[12]), TWING[13, 14], TLNS3D[15], and OVERFLOW[16]. These flow solvers provide a range of physics from linear potential, to full potential, to Navier-Stokes flow analysis and can be used for a range of configuration geometry from wing alone to wing-body-tail. A502 provides linear potential analysis, TWING provides full potential analysis and TLNS3D and OVERFLOW provide Navier-Stokes flow analysis. OVERFLOW and A502 can accommodate up to wing-body-tail configurations, TLNS3D wing-body and TWING wing alone. Data from the flow solvers is stored in the DMF database. This information is then used to support aerodynamic optimization directly, without augmentation from other utilities, or is used for performance-based optimization. In the latter case, the aerodynamic data generated by the flow solvers is augmented in several ways depending on user options to create drag polar tables (lift vs. drag) for use by the performance code, Boeing's FLEXMIS. Since the performance code expects trimmed data, two options have been provided to generate stability derivatives (change of lift, drag and pitching moment with respect to angle-of-attack and control surface deflection) to support trim polar calculations.

The aerodynamics domain process was developed to provide aerodynamic estimates from a variety of flow solvers. The basic process illustrated in Figure 10 is independent of the flow solver selected, where scripts control the perturbation of the seed geometry, design constraint checking, generation of single-block or overset grids, creation of input and execution of the selected flow solver, and the development of the aero database and post-processing. While the same basic steps are taken for the flow solvers each has specific and tailored processes to accommodate individual solver requirements. The process template used for get-all-flow-solutions was used to generate flow solutions at angles-of-attack in addition to basic design conditions and to implement aeroelastic processing. Combinations of the linear and non-linear flow solvers can be invoked to generate flow derivatives more efficiently and/or to enlarge the number of points developed for CD vs. CL performance module input tables. As part of the aerodynamic process a tool (OVERSENS) is in development to provide OVERFLOW chimera grid sensitivities for flow derivatives[17].

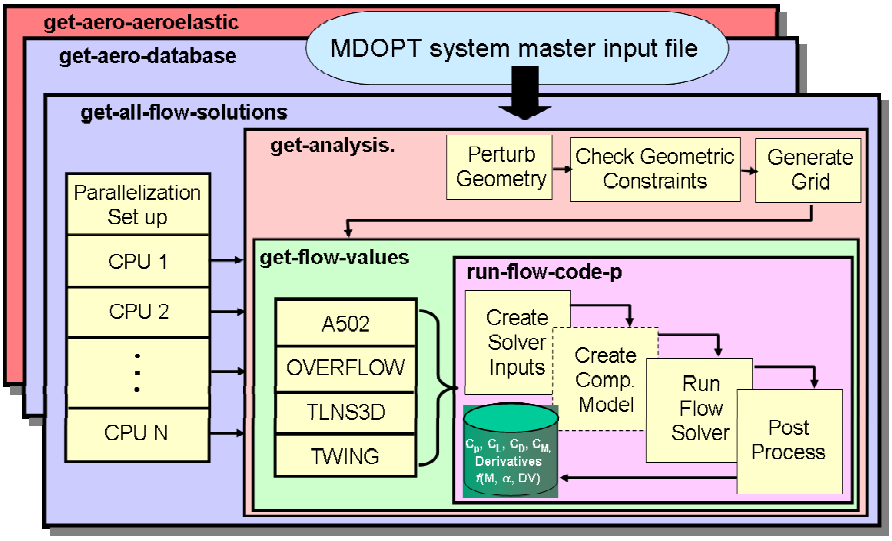

Figure 10 Aerodynamic process control

Structures

MDOPT supports two tools for the structural weight optimization and cg determination. An overview of the optimization process for computing air vehicle weights is displayed in Figure 11. The Boeing SeaTac code (Synthesis and Evaluation of Advanced Tactical Aircraft Concepts[18]) is used to generate a preliminary design level estimates of weight and center of gravity (cg), while ELAPS[19, 20] is available to refine the initial SeaTac estimate. SeaTac is a preliminary design tool for the rapid sizing and evaluation of aircraft. It includes the necessary system considerations required to do this including aerodynamics, propulsion and weights, including subsystem weights. The weights module(s) are an eclectic collection of empirical relationships that cover all configurations placements and weights. In the MDOPT environment, the user has the option of using SeaTac structural weights or updating the SeaTac weights by invoking the SAB ELAPS optimization process [21]. The ELAPS code is a design-oriented structural analysis tool developed at the NASA Langley Research Center. It is intended to provide high-fidelity analysis at low modeling and computational cost. It is based upon equivalent plate theory. The SAB ELAPS process as implemented in the MDOPT system is a collection of FORTRAN programs and UNIX scripts that govern and control the analysis for a given design point configuration. Optimum structural weight and the structural characteristics (mass and stiffness matrices, inertia tensor) of an elastic aircraft are the results. The MDOPT ELAPS process performs the following functions for each design point (subcase) in the global design space:

The core of the MDOPT ELAPS structures process is the SAB system developed at NASA Langley, consisting of programs that perform the aeroelastic structural optimization (aeroelastically converged minimum weight design) including equivalent plate analysis and sensitivity computation (ELAPS),

aerodynamic load computation (UDP), flutter analysis and sensitivity computation (DLFLUT), and trim and structural optimization (KSOPT). Furthermore, SAB calls the aerodynamic analysis program (lifting surface method), flutter analysis and derivative (unsteady aerodynamics - doublet lattice for subsonic), the optimization set up program, the optimizer (KSOPT) and the data base program. A good overview of the SAB ELAPS optimization process can be found in reference 22.

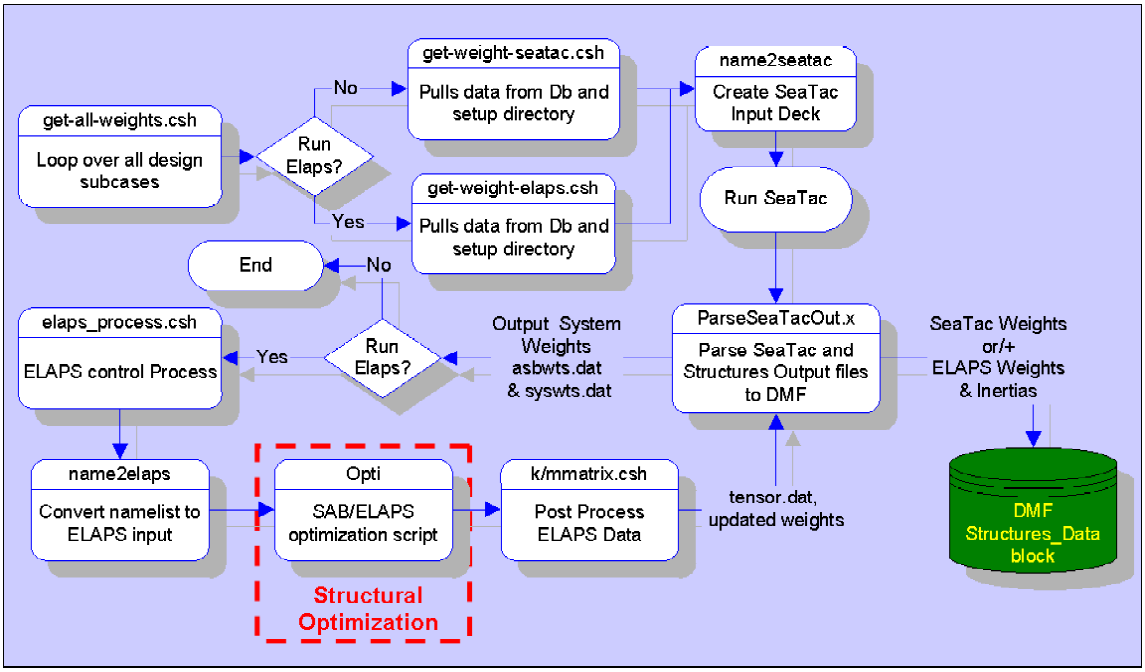

Figure 11 Structures weight build up process.

Stability and Controls

In the initial phase of MDOPT development feasibility of the S&C domain was demonstrated with a somewhat simplified implementation. Acknowledging that aircraft stability requires three-axis control, the development was limited to the longitudinal, or pitch-plane, axis. This implementation is based on a multi-disciplined approach that requires interactions with geometry, aerodynamics, flight controls, weight, inertia, and propulsion as outline in Figure 12. The longitudinal control constraints, designated by military specifications, Federal Aeronautics Administration (FAA) regulations, or unique mission requirements, are stored in the database, DMF. The longitudinal characteristics are analyzed by rapidly solving the equations of motion and, thus, can be queried directly by the MDOM as needed. The equations of motion are derived generically, are applicable for all geometric configurations, and are valid for any range of fidelity of inputs, i.e., linear and nonlinear aerodynamics, and flexible structures. Three basic S&C computations are made for trim, pitch control as defined by a user input roll rate and nose wheel lift off.

The trim routine determines the angle of attack and control surface deflection required to trim the air vehicle at an arbitrary flight path angle and load factor. Additionally, there is another computed parameter which is the total required horizontal deflection to meet aircraft trim and the additional longitudinal control required in roll. The aerodynamic modeling for trim includes the standard aerodynamic effects of lift, drag, and pitching moment. Propulsion terms include gross thrust and ram drag, while mass properties contributions include weight and center of gravity location. Finally, the trim routine takes into account the flight trajectory which is modeled as functions of load factor, flight path angle, and airspeed.

Along with the trim calculation, MDOPT computes the additional pitch control required for roll. This additional pitch control is added to the trim control requirement to derive the total horizontal tail deflection requirement.

Like the trim routine, the nose wheel liftoff routine is also a generic set of equations. NWLO occurs when the load on the nose gear goes to zero during takeoff. Given the allotted pitch controller deflection, this module will solve for the takeoff speed or most forward allowable cg. There are two modes, first the

takeoff velocity is computed for a given center of gravity location, while the second mode will compute the most forward allowable center of gravity for a speed. Nose wheel liftoff parameters (cg or Ve) are calculated based upon the user defined horizontal tail deflection, takeoff flight parameters, and full non-afterburning thrust applied.

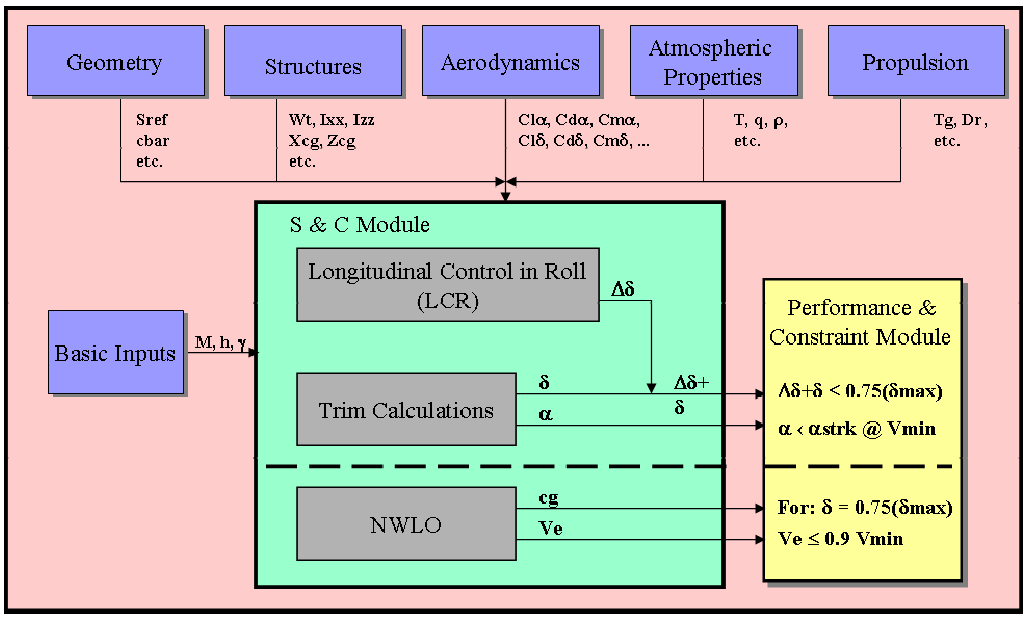

Figure 12 S&C process and I/O

V. MDO Manager (MDOM)

The MDOM is used to set up the global optimization problem, provide the initial input data for the aerodynamics, structures, and S&C domains to enable their respective analyses and potential sub-domain optimizations, and to control aspects of those domains that have been sufficiently automated. In addition, it controls the global MDO process and provides functional utilities in the form of services for constituent elements of the process, e.g., geometry management or response surface definition. As outlined in Figure 1 the MDOM consist of several major elements.

Performance Constraint Module: This module supports the evaluation of global-level objective and constraint functions e.g. range. The mission performance code FLEXMIS[23] was integrated as the performance constraint model for MDOPT. The FLEXMIS code is a modular mission performance tool that allows mission profiles to be built up from combinations of elementary missions segments. Vehicle characteristics are input to the code, and the range, radius, or endurance performance consistent with these characteristics and the specified mission profile are calculated. The code provides a great deal of flexibility in the mission profile definition. FLEXMIS was well suited to the requirements of MDOPT. The required set of vehicle characteristics for input to FLEXMIS are within the capability of MDOPT, consisting primarily of planform and overall vehicle geometric data, trimmed drag polars, and propulsion information (normally read from external flat files). The input formats are compatible with MDOPT and can be set up using the system GUI. FLEXMIS computes mission performance using 3DOF equations of motion.

Response Surface Generation Facility: One of the final by-products from each of the individual domains will be data needed to form response surfaces. This data will be stored in the DMF database, retrieved and assembled into a response surface by the Response Surface Generation Facility within the MDOM. Response surfaces or surrogate models are approximations to more detailed and expensive discipline computer analysis programs. These models may be least squares models, or in the case of the MDOPT system, models that interpolate. In either case they are constructed to approximate the exact objective

function surface by sampling design space at a carefully selected number of parameter value settings, or sample points. The surrogate model is constructed by running the expensive analysis code at the sample points and then fitting the surface. Surrogate models provide the designer with insight into the behavior of a complex optimization problem by allowing the user to identify and eliminate variables that have little effect. The function optimizer then interrogates the surrogate model for objective function values rather than run an expensive analysis code thereby reducing the total number of function evaluations for the optimization process.

MDOPT contains the suite of software known as DACEPAC[24] developed by Andrew Booker. This technology library is comprised of the three main elements, experiment generation, response surface model fit and model interrogation-evaluation. Design and Analysis of Computer Experiments (DACE[25]) is a method of exploring the relationship between a computer simulation's input variables and its output values. This exploration has three elements. First, the establishment of a judicious choice of combinations of settings of the input variables at which to run the simulation and obtain output, or response, values. Second the creation of a model of the responses as a function of the input variables, known as a surrogate model or interpolating response surface (IRS). The third aspect is interrogation of the model and analysis of the experiment in terms of the individual and pair-wise effects of the variables on the IRS and in terms of the importance of these effects. MDOPT implementation of DACE provides a systematic and automated way to cover the input space when sampling output. Distributions of sampling points throughout the design space can be obtained by one of two methods. First, optimal experimental designs based on a probabilistic interpretation of a model fit[25] are available for cases with 15 to 20 design variables. The other space filling design capability is orthogonal arrays in the sense of Owen[26]. The resulting kriging model (IRS) interpolates the responses into a fit based on an assumption of an underlying Gaussian process. This model provides a global look at the input-output relationship. DACE analysis of this IRS model can then be used to find good regions of input space for further exploration or optimization, gain insight into regions with candidate optima, reduce the number of input variables, and summarize the results of a DACE experiment through graphical presentation.

Geometry Management Facility: This facility was built upon the David Taylor Non-Uniform Rational B-Spline (DTNURBS) geometry library[27]. The primary functions are to provide geometry management including geometry input/output, perturbation and constraint evaluation. Available input options include IGES surfaces, parametric lofting, or user defined surface grids in ASCII or PLOT3D formats. An extensive library of perturbation and constraint options are included for wing, body, tail and general shaping as listed in Figure 13. The options provide a wide range of perturbation options for the user to choose from, in addition to numerous constraint options. A major enhancement to the standard set of design variables in the MDOPT system has been the solid options which give the user the ability to wrap existing geometries with solids allowing users to input surface grids without requiring field grids to be recreated for each perturbation. The idea is somewhat based on the use of solids or free form deformations[28] in the software animation industry. MDOPT uses solids of NURBS based splines. The user can define design variables as a subset of the solid knots to be moved in any of the coordinate directions or in a direction normal to a given solid face (e.g. normal to a body surface).

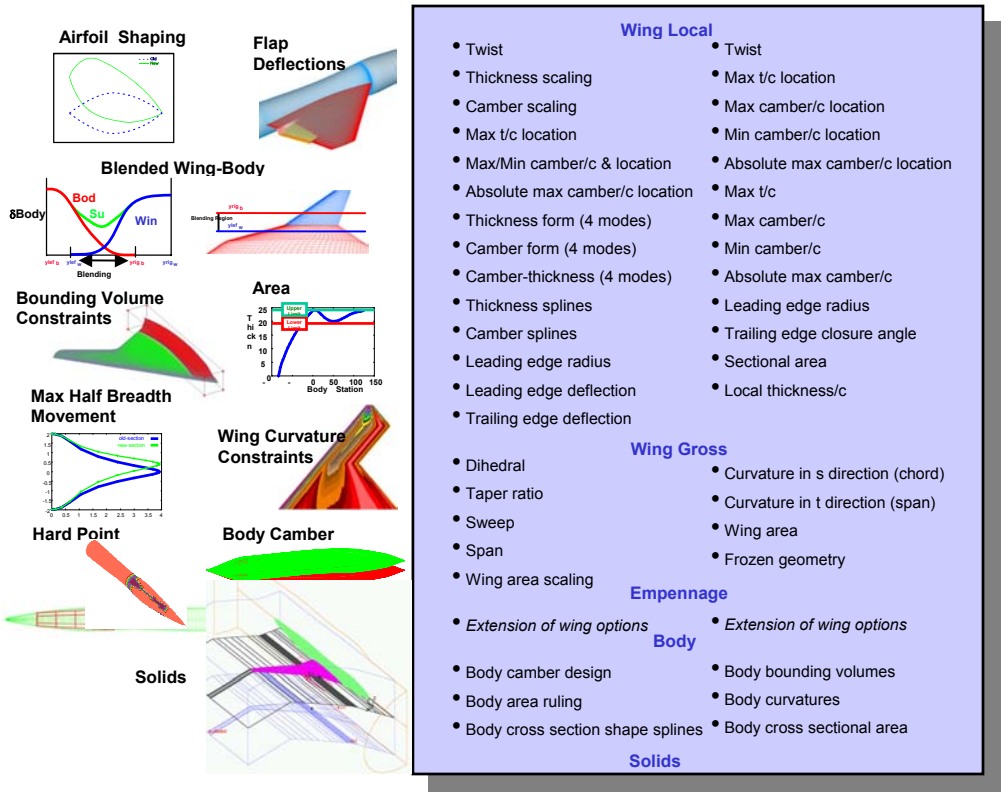

Figure 13 MDOPT geometric design variable and constraint options.

Optimization Module: The optimization module consists of a library of optimization approaches and system integration interfaces. This library contains both local and global optimization search tools that are user-selectable from the MDOPT GUI. Identifying the settings of design variables that provide improved performance is accomplished using one of several optimization strategies included in the system. There are two general categories of optimization strategies provided within the MDOPT system. A global design space search with function calls made to surrogate models (IRS) and local design space search with function calls made directly (Direct) to the analysis modules.

Each of the optimization tools has an associated list of input and control parameters that influence their performance. These parameters have all been set to predetermined values to provide general, overall robust performance. Each method is performing a constrained global search of the user specified design space to solve the mathematical problem that is the minimization of a scalar function of design variables subject to constraints.

Minimize:	$f(x_i)$	$i = 1, n$
Subject to:	$c_l \leq g(x_i) \leq c_u$	$i = 1,m$
Within the range:	$x_l \leq x_i \leq x_u$	$i = 1,n$

where $f(x_i)$ is the objective function of a vector x of n design variables, $g(x_i)$ an m vector constraint function with upper and lower bounds , c_l and c_u, and x_l and x_u are the bounds on the design variables defining the design space. Interrogating an approximate model of the objective function over the design space completes the optimization search. Aerodynamic constraints are also modeled with an IRS. Geometric constraints are imposed on the problem with the use of penalty functions or direct geometric evaluations. Penalty functions for geometric constrains were a necessity for the genetic approach and are also applied with the NPSOL based approach to reduce optimization search times. NPSOL[29] is a gradient optimization

program developed by Stanford University and the genetic approach is a Boeing developed hybrid genetic-gradient optimization routine.

Global optimization with NPSOL uses a strategy that identifies random starting points in the design space from which to initiate a local, gradient search. N random points are selected within the design space. Each of these points is checked for proximity to the other points in the sample. Any points selected that are within a specified distance are rejected and reselected. From each of the individual points separate constrained NPSOL searches are initiated and the resulting design variable settings and objective function values collected for further evaluation. This list of settings is then searched for the minimum value of the objective function thereby identifying the global optimum. The probability of correctly identifying the global optimum is directly proportional to the number of random points, N. Currently N is set to 50 in MDOPT.

Global optimization with a hybrid method consisting of a genetic algorithm (GA)[30] and a sequential quadratic programming (SQP)[31] [32] is also available as a driver. The GA is used to initiate the optimization and locate the region of the global minimum point. At a pre-selected switch point, the SQP method begins with the best candidate found by the GA and locates, with high accuracy, the global minimum point.

Response surface model refinement and re-optimization can be performed to further improve the global design space search using several methods. Examples are displayed in Figure 14, which includes four options: Approach 1 - Simple iteration, of create a model from an n sample size, find the optima from the model, run confirmation flow solutions at the optima, add then these runs into the design space data base and repeat Approach 2 - Model refinement using an engineer in the loop, basing refinement decisions on available model characteristics, i.e., design variable main effects and interactions, to restrict the range of design variables to areas of design improvement as identified by the current model. Approach 3 – Stratified balanced local-global search (SBLGS) model refinement method, where points are systematically added locally near a model optimum and globally at points throughout the design space where model errors are large. Approach 4 - Refine approximate global model predicted design with local gradient driven approach with starting points defined by observed optima from the approximate model

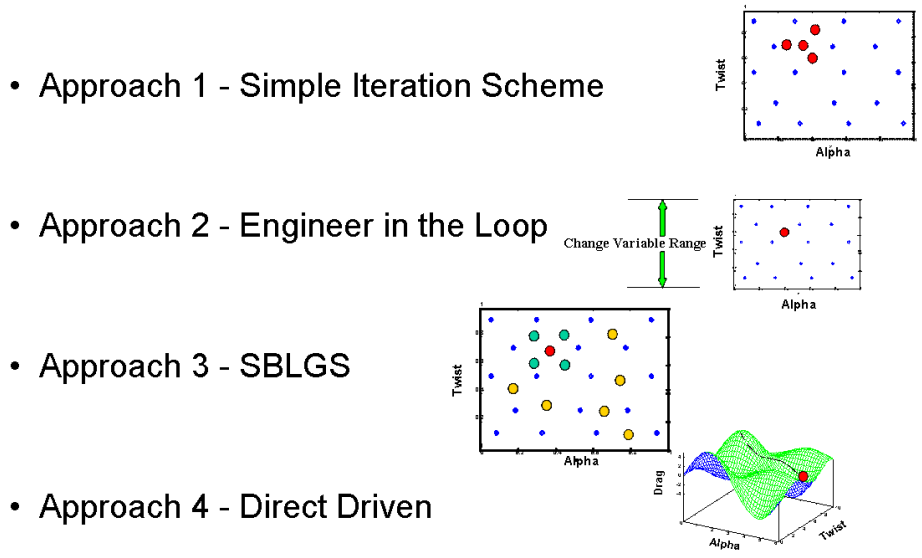

Figure 14 Optimization refinement schemes.

Option four, the direct driven optimization method, is available within the system. This method is a calculus-based approach using NPSOL. The NPSOL driver program invokes a sequential quadratic programming (SQP) method to solve the optimization, or nonlinear programming problem, described above. The direct driven optimizer has been set up for use in three modes, aerodynamic optimization with finite difference derivatives, aerodynamic optimization with analytic derivatives and performance-based optimization with finite difference derivatives. All three approaches share the benefits and limitations common to calculus-based methods. Their design space search is a local, as opposed to a global strategy. The results will be sensitive to the starting point used. i.e., in a design space where multiple optima exist,

they will stop at the nearest optimum to the starting point. They are sensitive to discontinuities in the design space. As such, the optimization process may need to be restarted from interim points. They require calculation of objective and constraint function sensitivities (derivatives) to changes in design variables. For expensive function evaluations, e.g., CFD flow solves, a single optimization run may require computer resources on the order of that required for a global optimization run. However, with the gradient-based type of optimizer directly connected to the function evaluator, it usually can locate a local optimum with greater fidelity than either of the global search techniques within MDOPT provided accurate sensitivities are obtained

Problem Definition Facility: This component is accessed through the GUI and supports the user's specification of the global design variables, constraints and objective functions, and control information to guide the optimization process.

Graphical User Interface (GUI): All user interaction within MDOPT is supported by an easy-to-use graphical user interface. With only a few exceptions (e.g. initial ELAPS plate model build up) all user input is created through the GUI. The GUI is laid out in a tabbed notebook format as displayed in Figure 15, where the user follows the tabs from left to right filling out input in the order of task performed. Tasks are submitted to machines defined by the user across the network using a Run pull down menu as displayed.

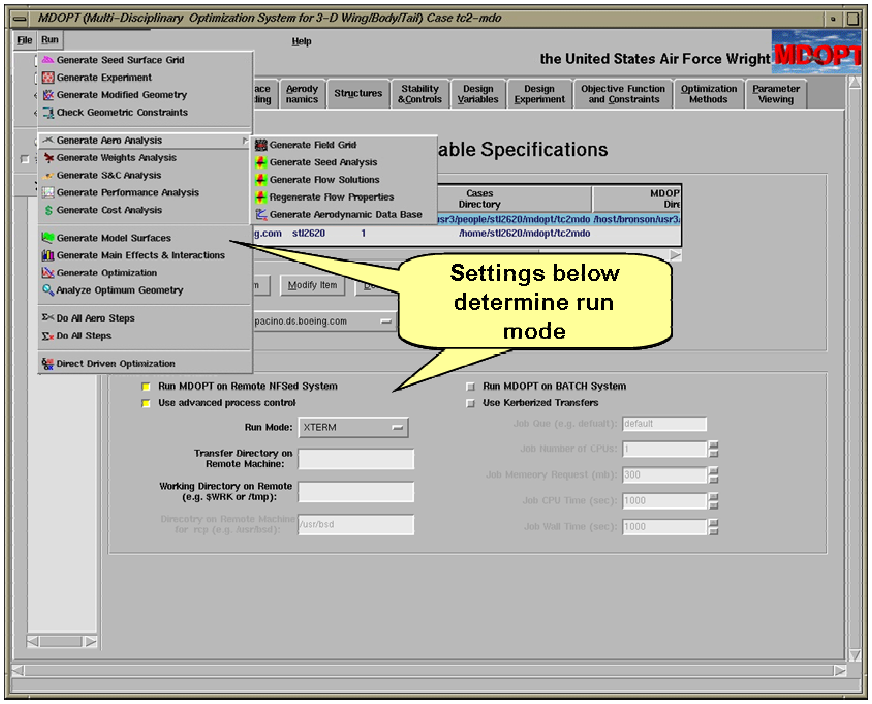

Figure 15 Example GUI page show tabbed notebook layout

Output Generator: The output generator enables the user to view the optimized geometry and other features of the design space upon completion of the optimization process. The geometry viewer supports visualization of wireframe and shaded graphics representations. It also supports interactive selection and display of desired geometry characteristics and design variable distributions and constraints, in addition to 'canned' plots. A sample of plots created can be seen in Figure 16, which displays airfoil plots of perturbed geometry using the MDOPT GUI with GNUPLOT[33] and a 3D viewing widget for grids, based on the OVERGRID og TCL/TK widget developed by William Chan[34]. Any design point within the experiment can be selected for plotting or combinations of designs can be simultaneously plotted for inspection.

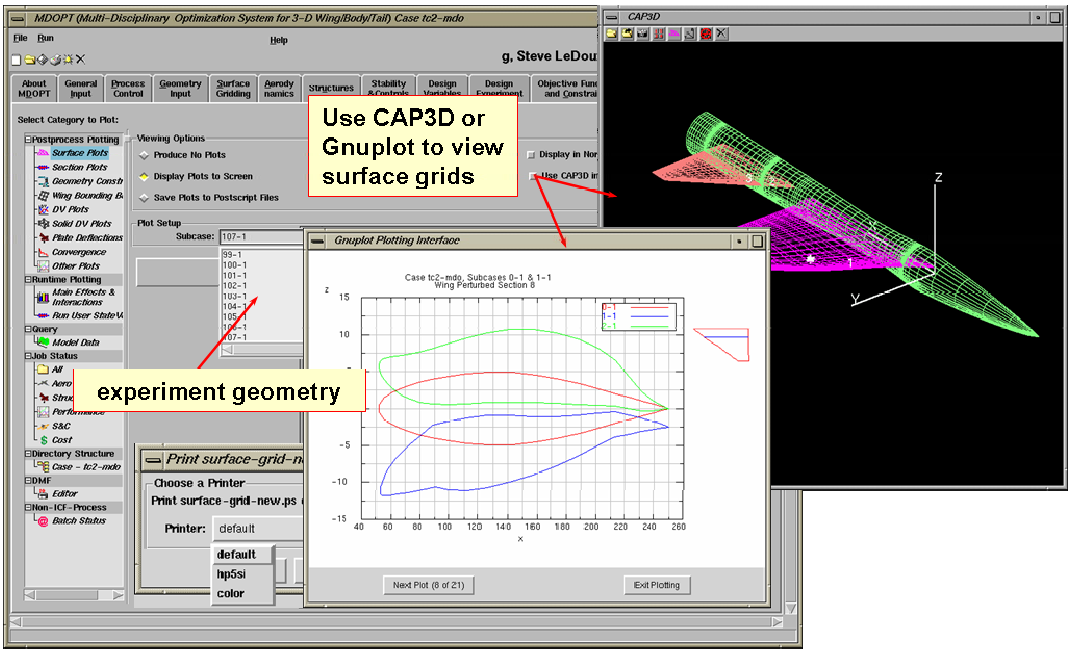

Figure 16 Output Parameter Viewing

VI. Test Cases

MDOPT system testing included two sample application problems, one using lower order codes and the other using complex computational models in conjunction with lower order tools. Only the test case based on higher order analysis tools, designated System Test 2, will be discussed in this paper. The optimization problem, described in Figure 17, consisted of maximizing range for a fixed mission profile of high altitude cruise, low altitude penetration and weapons delivery, and a return high altitude cruise. Aerodynamic data was supplied through A502 (panel code) and OVERFLOW (Navier-Stokes) and SeaTac (preliminary design sizing tool). Controls data was derived from implemented trim and longitudinal control in roll modules from ASCAM (Aerodynamic Stability and Control Analysis Module). Structures data was obtained using a combination of SeaTac for fixed system component weights, e.g., landing gear, avionics, etc., and ELAPS (Equivalent Plate Solution computer program from NASA Langley) for structural sizing and weight of main aircraft components, wing, body, and horizontal tail. Mission performance was determined using FLEXMIS (flexible mission analysis program).

Performance inputs to the mission analysis tool required trimmed drag polars in the form of tables of lift versus drag. To accomplish this, the system called on a combination of flow solvers cited above where A502 was used to generate stability and control coefficients, e.g., $C_{L\alpha}$ (lift curve slope) and $C_{M\delta}$ (control surface effectiveness). These were used to trim the OVERFLOW generated lift, drag, and pitching moment for the basic wing/body/tail configuration and to supplement the drag polar curve fits. Two or four Navier-Stokes OVERFLOW solutions were completed for each of several Mach numbers spanning mission flight conditions. Curve fits were created through the data sets at each Mach number represented. These were, in turn, evaluated to fill out the required tabular input for the performance code. All of which is completed automatically by the system.

Objective:

- maximize range for a specified mission profile
- linear physics plus SEATAC weights

Mission:

- Cruise outbound (transonic)
- Penetration/egress (high transonic) - deliver payload
- Cruise inbound (transonic)

Constraints:

- Max TO weight
- Fixed span
- S&C performance requirement
- No body changes
- CL as function of Mach - fixed payload
- Fixed SFC schedule

Design Variables:

- Wing: max t/c location (3)
- Wing: twist (3)
- Wing: camber (3x4)
- Horizontal: taper ratio (1)
- Horizontal: area scale (1)

TOTAL = 20 DVs

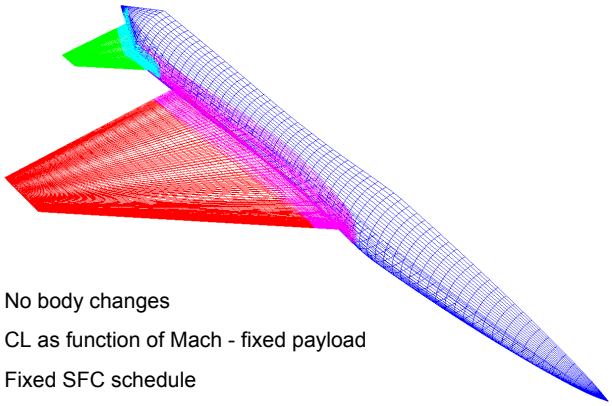

Figure 17 Test Case Problem Setup

The optimization process follows the same overall approach outlined in Figure 2. A DACE (design and analysis of computer experiments) orthogonal array experiment is generated defining a set or population of unique geometric configurations representing the range of design variables. Aerodynamic, structures, controls and performance data is generated for each configuration. Approximate models are then constructed of objective and constraint functions. The optimization is then performed from these approximate models. Details of this process are shown in Figure 18.

The values of objective function, mission range, obtained for each site in the 125 site experiment for each drag polar model are plotted in Figure 19. This is a simple means of manually reviewing the characteristics of the data base. A quick look allows identification of the amount of variation in the data base, areas of potential "good" designs and areas where there may be some computational problems. This plot and the additional bar plots detailing variation of domain discipline data are not automated products produced by the system but can be created manually with a few simple steps from system generated data. Table 1 describes the results of four separate system applications to the optimization problem using two or four points to define a polar with and without derivative augmentation of the polar curve fit. The eight point polar data represents analysis completed to confirm optimization results. Eight Navier-Stokes solutions were completed for each Mach number considered using the optimal geometry derived from the four approaches. With this higher data density it was possible to evaluate the relative error of the coarser approaches. It is clear that error is significantly reduced with additional data points. However, only the four point polar approaches produced a design with improved performance as compared to the baseline value of 804 nmi.

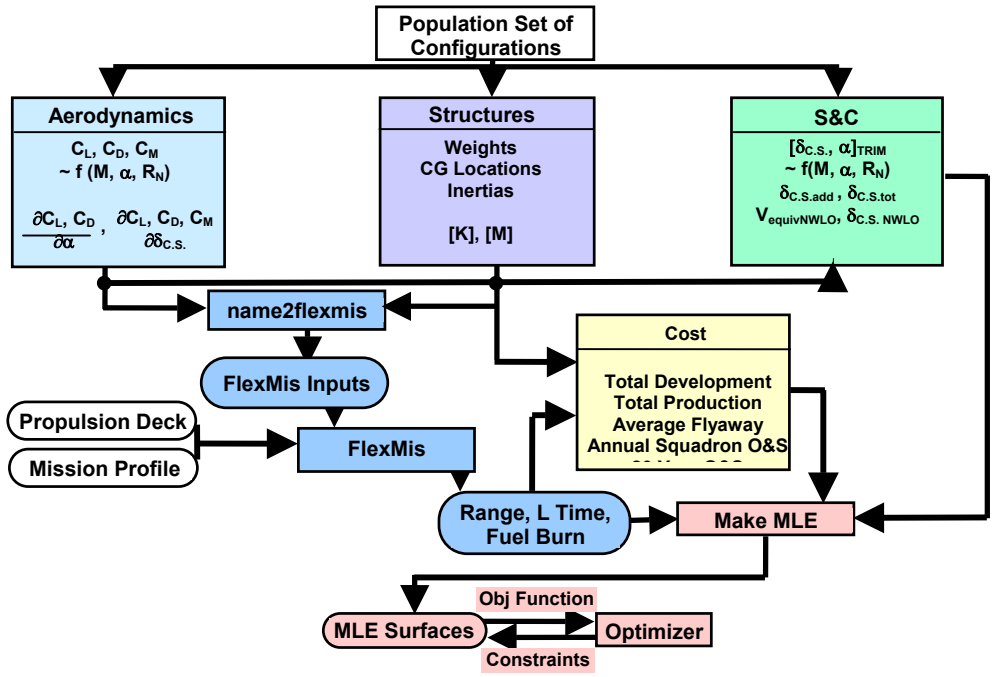

Figure 18 Global Optimization Process

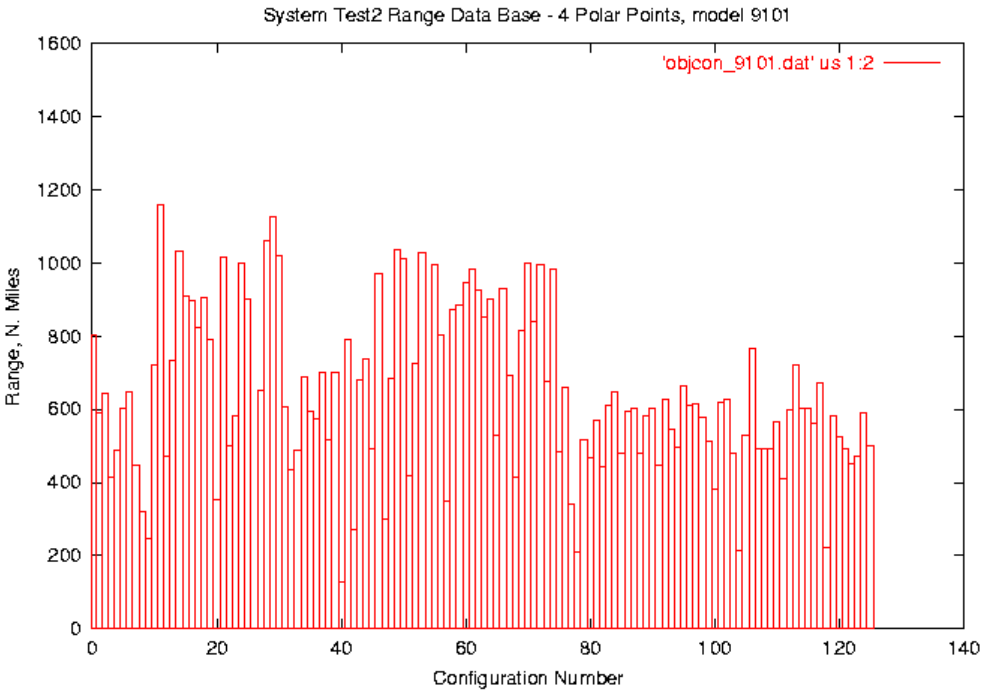

Figure 19 Database Variation of Range

Table 1 Optimization Results

	2 Point Polar, No Derivatives		4 Point Polar, No Derivatives		2 Point Polar, With Derivatives		4 Point Polar, With Derivatives	
	Original	8 Point Polar	Original	8 Point Polar	Original	8 Point Polar	Original	8 Point Polar
Range, N. Mi.	1229.	599.	842.	850.	819.	720.	882.	822.
Elapsed Time, Min	301.91	164.76	217.81	218.45		214.24		216.34
Total Fuel, Lbs	6241.7	6241.4	6238.9	6238.5		6085.2		6071.7
Reserve Fuel, Lbs	329.3	328.6	326.1	326.5		317.8		319.3

VII. Conclusions

The MDOPT system represents a major step forward in the development of design optimization capability. It is a unique combination of higher order geometric functionality with Navier-Stokes level flow analysis, approximate modeling techniques and robust numerical optimization schemes in an automated framework that provides an extensible method for inclusion of multiple discipline domains. The system provides a means to achieve increased air vehicle performance while controlling computational resources during a design cycle. Flexibility of use and potential for growth has been extended through a modular system design that will run on high performance computers as well as engineering workstations.

VIII. Acknowledgements

MDOPT was developed under joint funding from the Air Force Research Laboratory, Wright-Patterson AFB, Ohio, under the Multi-Disciplinary Optimization Using Computational Fluid Dynamics, MDOPT, Contract Number F33615-98-2-3014, and from Boeing cost match funds. The work was performed by the Boeing Company, Phantom Works Organization, in Seattle. Dr. Don Kinsey and Lt Charles Hoke were the USAF Project Engineers. Gasper Fatta was the Boeing Program Manager and Dr. William Herling was the Principal Investigator. In addition to the authors, program development was completed by Mr. Dean Barron, Mr. Gordon Blom, Dr. Andrew Booker, Mr. Kwan Chang, Mr. Dr. Jonathan Elliot, Ranald Engelbeck, Dr. Paul Frank, Dr. Neal Mosbarger, Mr. David Treiber, and Dr. Matt Warfield. Contributions from Lt. Charles Hoke are gratefully acknowledged. His testing and feedback during the development of this system was a key component to the successful completion of MDOPT. Finally, acknowledgment is given to NASA for the technology contributions to several of the system modules. These include OVERFLOW, Chimera Grid Tools, CSCMDO, TWING, TLNS3D, HYPGEN and WINGDES.

IX. References

[1] Tcl/Tk Developers website, http://www.tcl.tk/ [cited May 2004]
[2] Mico **M**ICO **I**s **CO**RBA website, http://www.mico.org [cited May 2004]
[3] Combat website, http://www.fpx.de/Combat/ [cited May 2004]
[4] CORBA 2.3.1 Specification, http://www.omg.org/cgi-bin/doc?formal/99-10-07 [cited May 2004]
[5] MySQL website, http://www.mysql.com [cited May 2004]
[6] Herling, W.W., Emsley, H.T., LeDoux, S.T., Ratcliff, R.R., Treiber, D.A., Warfield, M.J., "3DOPT - An Integrated System for Aerodynamic Design Optimization", AIAA Paper 98-2514, June 1998, 16th Applied Aerodynamics Conference, Albuquerque, NM.
[7] Portable Batch System, http://www.openpbs.org, Altair Engineering, Inc, [cited June 2004]
[8] CORBA Product Matrix Comparison, http://www.puder.org/corba/matrix [cited May 2004]
[9] CORBA Event Service 1.1 Specification, http://www.omg.org/technology/documents/formal/event_service.htm [cited May 2004]

[10] Tcom website, http://www.vex.net/~cthuang/tcom/ [cited May 2004]
[11] MySQLTcl (Tcl bindings for MySQL) Website, http://www.xdobry.de/mysqltcl/ [cited May 2004]
[12] Johnson, F., "A General Panel Method for the Analysis and Design of Arbitrary Configurations in Incompressible Flows", NASA CR 3079, May, 1980
[13] Holst, T. L., "Fast, Conservative Algorithm for Solving the Transonic Full-Potential Equation", AIAA Paper 79-1456, July 1990.
[14] Thomas, S. D., "TWING - A Transonic Wing Analysis Computer Program", Contract NAS 2-11555/917, November 1990.
[15] Vatsa, V. N., "Accurate Numerical Solutions for Transonic Viscous Flow Over Finite Wings," *Journal of Aircraft*, Vol. 24, pp. 377-385, June 1987.
[16] Buning, P.G., et al, "OVERLFOW User's Manual, Version 1.8j", NASA Langley Research Center, February 1999
[17] Elliot, J. K., Herling W. W., "Chimera Sensitivity Project, A Chimera Approach to Aerodynamic Shape Optimization for the Compressible High-Re Navier-Stokes Equations", 8th AIAA/NASA/USAF/ISSMO Symposium on MDO, September 6, 2000, Long Beach, CA
[18] Engelbeck, R. M., Gronenthal, E. W., "SeaTac V1.1 User's Notes", Boeing Document D950-10416-1
[19] Giles, G. L., "Equivalent Plate Analysis of Aircraft Wing Box Structures with General Planform Geometry", J. of Aircraft Vol. 23, No.11, 1986.
[20] Giles, G. L., "Further Generalization of an Equivalent Plate Representation for Aircraft Structural Analysis", J. of Aircraft Vol. 26, No. 1, 1989.
[21] Wrenn, G. A., "NASA/SAB Optimization Process Description (Draft)", NASA Langley Research Center, Hampton, VA."
[22] Sexstone, M. G., "Aircraft Structural Mass Property Prediction Using Conceptual-Level Structural Analysis", SAWE 2410, May 18-20, 1998.
[23] Barron, D. A., "User's Guide to Performance Calculation Programs: FLEXMIS, PSPLOT, TOPDOG, and LEVEL", Boeing Document D658-10615-1
[24] Booker, A. J., "Design and Analysis of Computer Experiments", AIAA Paper 98-4757, September 1998, 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization
[25] Sacks, J., Welch, W. J., Mitchell, T. J. and Wynn, H. P., Design and Analysis of Computer Experiments, Statistical Science, 1989, Vol 4, No. 4, pages 409-435.
[26] Owen, A. B., Orthogonal Arrays for Computer Experiments, Integration and , Statistica Sinica, 1992, Vol 2, pages 439-452.
[27] Boeing Information & Support Services, "DT_NURBS Spline Geometry Subprogram Library Reference Manual Version 3.1," Naval Surface Warfare Center/Carderock Division, David Taylor Model Basin, November 1995.
[28] Sederberg, T.W. and S.R. Parry, "Free-form deformation of solid geometric models," Proceedings of SIGGRAPH '86 (Dallas, Tex., Aug. 18-22, 1986). In Computer Graphics, 20, 4, Aug. 1986, 151-160.
[29] Eldersveld, S. K., "Optimization Software Usage of NPSOL (Version 4.06) and LSSOL (Version 1.0)," Report MEA-LR-91, The Boeing Company, Seattle, Washington
[30] Holland, J. H., Genetic Algorithms, *Scientific America*, July 1992.
[31] Gill, P. E., Murray, W., and Wright, M. H., *Practical Optimization*, Academic Press, London and New York, 1984.
[32] Eldersveld, S. K., "Optimization Software Usage of NPSOL (Version 4.06) and LSSOL (Version 1.0)," Report MEA-LR-91, The Boeing Company, Seattle, Washington.
[33] Williams, T. W., and Kelley C., "gnuplot An Interactive Plotting Program," info-gnuplot@dartmouth.edu, 1996, http://www.gnuplot.info/, [cited May 2004]
[34] Chan, W. M., Meakin, R. L. and Potsdam, M. P.,"CHSSI Software for Geometrically Complex Unsteady Aerodynamic Applications", AIAA Paper 2001-0539, January, 2001