SAE Paper 1999-01-5578

Enabling Advanced Design Methods in an Internet-Capable Framework

Dr. Mark A. Hale and Dr. Dimitri N. Mavris
Georgia Institute of Technology

ABSTRACT

The enabling of advanced design methods in an internet-capable framework will be discussed in this paper. The resulting framework represents the next generation of design and analysis capability in which engineering decision-making can be done by geographically distributed team members. A new internet technology called the lean-server approach is introduced as a mechanism for granting Web browser access to frameworks and domain analyses. This approach has the underpinnings required to support these next generation frameworks – collaboratories. A historical perspective of design frameworks is discussed to provide an understanding of the design functionality that is expected from framework implementations to insure design technology advancement. Two research areas were identified as being important to the development of collaboratories: design portals and collaborative methods.

An internet-enabled design framework called IMAGE is highlighted and demonstrated using a probabilistic design example. The prototyped methods have found their way into a Conceptual Aerospace Systems Design and Analysis Toolkit used by the Air Force Research Laboratory.

EVOLUTION OF DESIGN FRAMEWORKS

A look first at the historical development of design frameworks before discussing their role on the internet is important in order to establish the functionality to be achieved by the frameworks. Any new framework developments or technology infusion must preserve the desired framework functionality. The premise for these frameworks is that they are to aid engineers in formulating, solving, and evaluating complex design problems with solutions that require analysis from several domains. Benefits of using these systems include design improvements and cycle time reduction. The automation of the solution process requires that control and communication of domain analyses be provided by the design framework.

Design Systems

The earliest design frameworks address the research question of *How can multidisciplinary vehicle synthesis be achieved using computers?* The framework requirements include:

- synthesis of several disciplines,
- autonomous simulation of problems, and
- usability/extensibility.

These frameworks utilize programming language constructs to achieve analysis integration. They are single executables in which control is exercised through algorithm design and the exchange of data (one component of communication) takes place through function or subroutine arguments and shared memory. This type of framework is commonly referred to as a "design system" and its arrangement is shown in Figure 1. The earliest of the systems are written in FORTRAN and a number of these systems are still in use today, including the FLight OPtimization System (FLOPS) developed by NASA Langley and AirCraft SYNThesis (ACSYNT) developed by NASA Ames.[1,2]

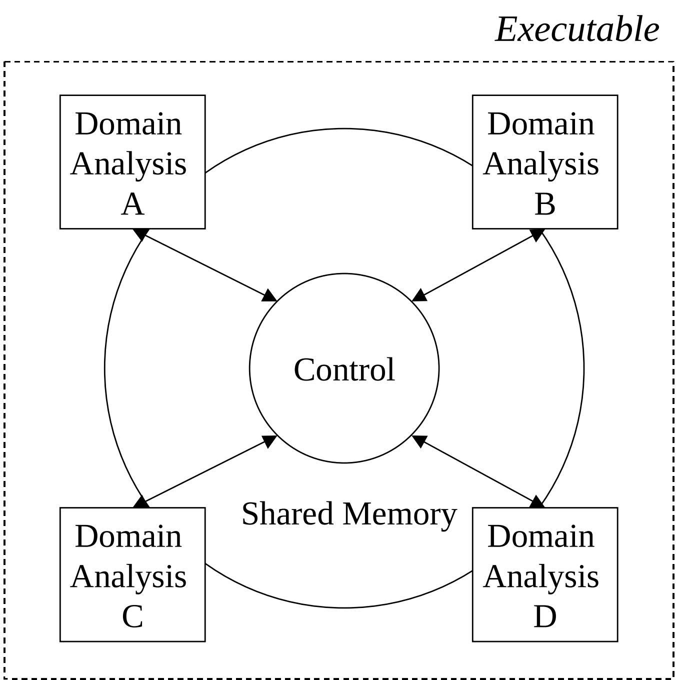

Figure 1. Design System Configuration

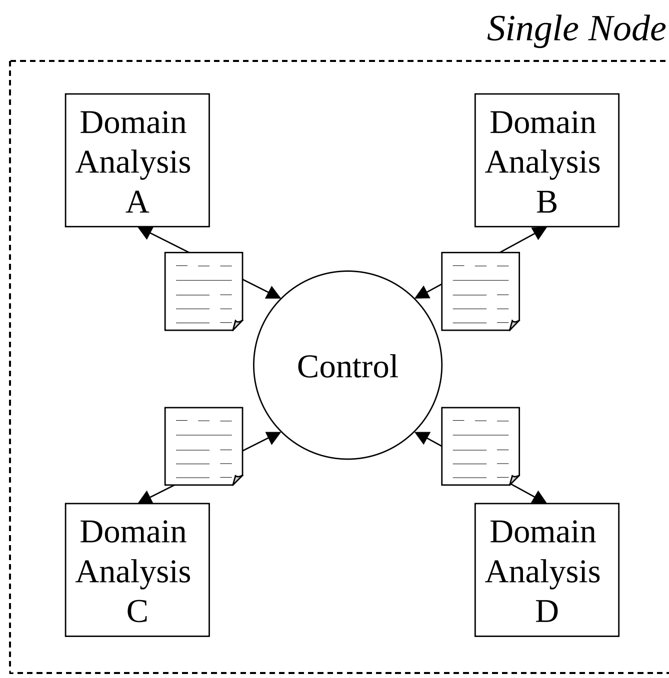

Figure 2. Design Framework Configuration

As a whole, these design systems provide tried and true rapid evaluation of configurations, trade studies, and system feasibility studies. However, fidelity developments in domain analyses easily outgrow their ability to be maintained in a single, integrated system. Often, systems are instead calibrated using high-order analysis tools. Cycle time reduction would be possible and more knowledge gained earlier[†] in the design process if higher-fidelity information and life-cycle disciplines could be included at the onset. This requires a re-configuration of analyses within the framework.

Design Frameworks

Frameworks thus evolved to address the question of *How can disparate, higher-fidelity disciplinary analyses be integrated for design and analysis?* The frameworks are required to provide:

- synthesis of high-fidelity disciplinary modules,
- accommodation of disparate analyses, and
- multi-analysis control.

There are several solutions to this problem. One is to use meta-models of the higher-fidelity analyses such as response surface equations and infuse them into design systems. Methods have been devised to automate this process.[3,4] At the same time, computer techniques emerged for multi-tasking and thus permitted the design framework orientation as shown in Figure 2. In these architectures, domain specific analyses are now self-contained executables and data is shared, typically in flat file format. The frameworks are typically operated on a single machine or in a strawman networked environment using remote shells and networked file systems. The Integrated Design and Analysis System (IDAS), the Design and Optimization Coupling Code (DOCC), and HiSAIR/Pathfinder are examples of this type of architecture.[5,6,7] (These frameworks offer additional capabilities not discussed here, such as optimization in DOCC, that add value to design processes.)

These frameworks provide straightforward methods for coupling analysis tools, resulting in quick problem solutions. However, scaling to accommodate problem complexity, whether from data fidelity or number of users, becomes difficult when these frameworks are used as part of a design enterprise. The difficulties arise because the frameworks rely heavily on translators, adding to the number of programs that must be maintained, and the inefficiency of flat files for storage of high-fidelity and dynamic information because of custom file formatting and revision control concerns.

[†] It is desired to have acquire as much knowledge as possible about a design when design freedom is at its highest and committed costs are at their lowest.

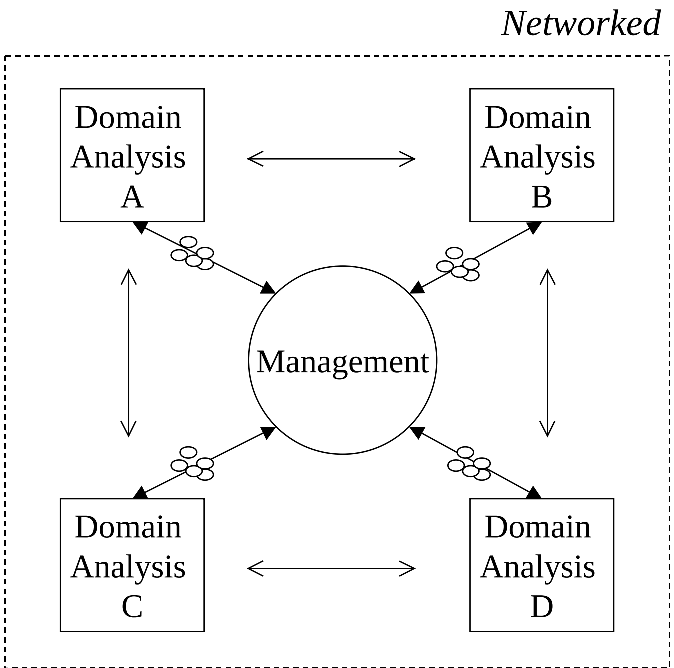

Figure 3. Heterogeneous Computing

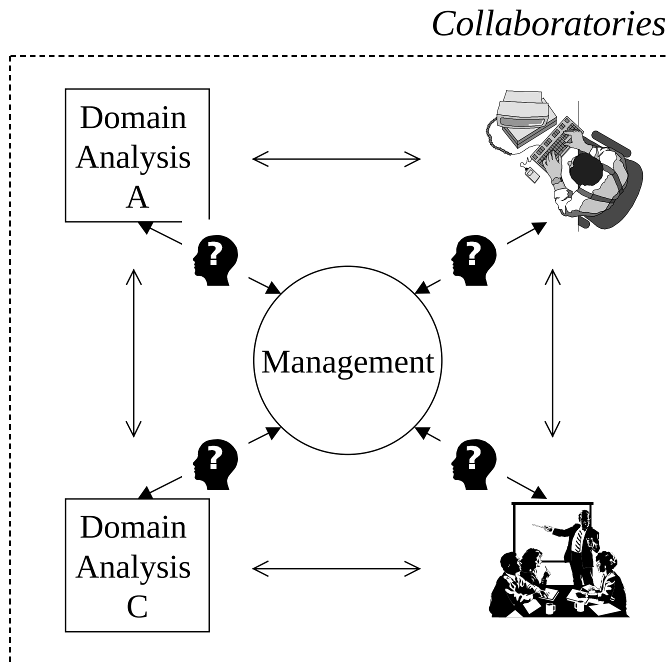

Figure 4. Collaboratories

Modern Design Frameworks

Modern frameworks examine the scalability issues of simulation in a corporate enterprise. The term "modern" is used here to describe those frameworks that incorporate recent research and technological advances that specifically address the use of frameworks in everyday design practices. They seek to provide an answer to the question *How can multidisciplinary frameworks be managed for the design of complex systems?* The framework requirements include:

- distributed decision-making,
- management of complex systems,
- communication of design information, and
- heterogeneity of data and resources.

These frameworks have a general architecture as shown in Figure 3. The domain analysis are executed in a distributed, heterogeneous environment. The analyses have the capability for peer-peer (analysis-analysis) transactions to facilitate high-end simulations. Messaging services utilize objects rather than using file-based exchange techniques. Management tasks now supercede the simpler control tasks present in earlier architectures.

The frameworks have matured to the point where their development has become profitable. Subsequently, there are a number of frameworks commercially available on the market, including ModelCenter™ (Phoenix Integration), iSIGHT (Engineous Software Inc.), RECIPE (International Space Systems Incorporated), and others. There are also a number of research frameworks as well, including Dakota (Sandia National Laboratories), Jane (Georgia Tech), and IMAGE (Georgia Tech).

Collaboratories

The frameworks discussed this far pull together information from disparate domain analyses and maintain an analysis structure in a user-transparent fashion. This leads us to the next line of reasoning: *What if the frameworks integrate elements from human decision-makers?* This is the emphasis of what is becoming known as a collaboratory. The notion of collaborative design conjures a grand vision of many designers, working at geographically distributed locations, making good design decisions in a reduced timeframe, realizing lower product cost, and producing revolutionary concepts never before thought possible. In fact, this is the vision of NASA's current Intelligent Synthesis Environment and the Department of Defense's Simulation Based Acquisition initiatives.[8, 9]

A collaborative framework is shown in Figure 4 in which computational simulations, teams, or teammates interact. Here, decision-makers coordinate in intermittently connected federations and when doing so exchange reasoning and experiences. This framework recognizes that not all decisions are made by a group of individuals collectively getting together (perhaps at geographically distributed locations through video conferencing techniques). In fact, designers will often apply lessons learned from these decision-making forums while executing analyses off-line. As an example, imagine the ability for a designer to automatically catalogue knowledge from a newsgroup and use the knowledge to steer analyses done off-line. This process also works in reverse; knowledge gained off-line can be documented and brought to the forum. The end goal is to give rise to better, faster, and cheaper designs through the communication of more than pure analytical results. Ideally, this relationship can be brought to fruition in a mixed human-machine environment in which decision-making occurs.

INTERNET CAPABILITY

The World-Wide-Web provides global internet services for information transactions and is an enabler for advancing the state-of-the-art in design frameworks. The successes of the Web can be attributed to affordable, common, platform independent interfaces that make computer transactions transparent. The Web consists of client and server applications operating on the internet as shown in Figure 5. In the client-server model, the number of clients is presumed to be far greater than the number of servers. HyperText Markup Language (HTML) documents are managed and distributed by Web servers using HyperText Transfer Protocol (HTTP). Client applications (browsers) load documents from the respective servers and make them available to the user. As a result, the user interface is logically separated from the information content, an important capability for geographic distribution. This capability is available through non-Web interfaces but most stand-alone applications have not taken advantage of the programming paradigms and the Web provides standards for doing so.

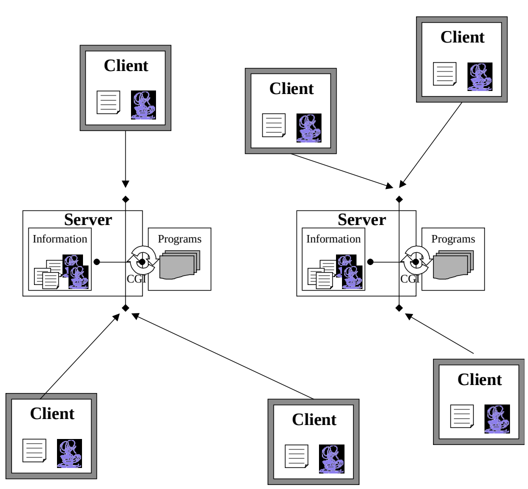

Figure 5. Internet Client-Server Configuration: The Status-Quo

The internet also has provisions for providing analysis capabilities that supplement static page material found on a server. Two methods are described next.

Common Gateway Interface

Servers have the ability to process application requests from the client using user-defined programs accessed via a Common Gateway Interface (CGI). This allows the server to make use of server-side processes for handling client requests. For example, information can be returned from the client page (for instance, from a form-based page) and processed by a server-side application operating through the CGI. The server then returns the results to the user.

Java

Java™ adds both client and server processing capabilities to the client-server model. These are called applets and servlets respectively and are depicted by the Sun Java™ logo in Figure 5. Applets can be run within Web browsers, as well as independent applications, and provide diverse functionality: including graphing, forms with local calculations, and animation with local rendering. These applets are downloaded from HTTP servers similar to the manner in which HTML documents themselves are downloaded. Java™ applets are stored and transferred in byte form to preserve source code but remain platform independent. JavaScript™, developed by Netscape Corporation, is similar to Java™ in that it adds client side processing. JavaScript™ is contained as text within HTML documents rather than transferred separately in byte form.

Applicability of Technology to Design Frameworks

There are several demonstrations involving the use of CGI scripts to construct the design frameworks as represented in Figure 2.[10,11] Java implementations have also been tested as aids in the design process.[12,13] These demonstrations have shown that it is possible to distribute analyses and their control using internet technology. In fact, they clearly show the benefits of separating the user interface from the information content and capability for geographic distribution.

A shortcoming of these technologies is that they do not push the current design framework envelope in addressing the functional requirements of enterprise management. An overview of specific complex systems design functionality of modern design frameworks was given in the historical perspective. This functionality must be preserved while harnessing the distributed user-interface capability made popular by the internet. This is important because this functionality is a precursor to collaboratories, the future of design frameworks.

LEAN-SERVER

The authors are investigating a novel server alternative for the internet, called a lean-server, as a mechanism for providing the functionality needed by design frameworks operating on the internet. A schematic of a lean-server is shown in Figure 6. The lean-server operates simply by a) receiving requests from an internet client, b) passing the requests to the software in which it is imbedded through the application's programming interface, and c) returning a response to the requesting client. The lean-server can be imbedded directly inside a domain analysis as shown in the figure or directly within a design framework (which manages its own set of analysis tools). Note that the domain analyses as used here may also be part of a peer-based simulation architecture. The benefits of this will be described in the following paragraphs. This gives flexibility in the manner in which the lean-servers are deployed. The lean server has added benefits of minimizing internet overhead, maximizing transaction speed, and insuring compatibility with client-side applications.[14]

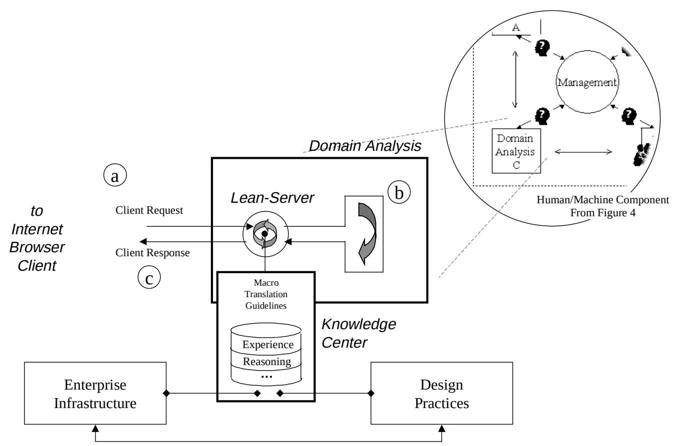

Figure 6. Lean-Server

Knowledge Center

Unique to this approach is that the lean-server has access to a knowledge center that can capture models of corporate practice through real-time learning and other knowledge processing techniques. These capabilities facilitate the functions of a collaboratory as described earlier by permitting consideration of both enterprise infrastructure and design practices and their influence in design decision-making. This capability is under further investigation as a gateway to providing solutions for intelligence-based frameworks, such as those required for NASA's ISE Initiative.[8]

Client-Server ⇔ Peer-Peer Bridge

The lean-server bridges peer-peer and client-server architectures by enabling internet requests to be serviced directly by simulation peers. The server is embodied directly with the simulation peer as represented in Figure 6 by a domain analysis. This enables the high-end inter-peer simulation activities to occur concurrently with Web requests. Simulation architectures such as the High Level Architecture consider timing, load-balancing of domain analyses, and fault-tolerance aspects not encountered in client-server architectures.[15] These aspects are important if frameworks are to include simulations incorporating high-fidelity domain analyses and enterprise models.

PRELUDE TO THE COLLABORATORY: ACCESS TO MODERN DESIGN FRAMEWORKS

The lean-server has been integrated into a modern design framework used for prototyping advanced design methods at the Georgia Tech Aerospace Systems Design Laboratory. The framework is called the Intelligent Multidisciplinary Aircraft Generation Environment (IMAGE).[16] As shown in Figure 7, IMAGE contains seven functional components indicative of common managerial components found in design frameworks to facilitate modeling, simulation, and design:

- *Application Programming Interface (API).* Underlying basic programming functions that are used throughout the system. Constructed around the shareware package tcl.[17]
- *Object-Oriented Database.* A design-oriented database that provides variable-fidelity objects, multiply-connected hierarchies, and accumulation of instances. Constructed around the shareware package tcl.[18]
- *Communications Library.* Routines that provide transparent communication among domain analyses operating in a heterogeneous network. Constructed around the shareware package PVM.[19]
- *Tool Manager.* User services for linking resources into the design system in a design independent manner. Includes domain analyses as well as user-interactive software.
- *Process Manager.* User services for decomposing design processes into manageable problems in which resources are linked sequentially or in parallel. Resources include state-of-the art design methods including design of experiments and probabilistic techniques.

- *Design Manager.* User services for populating design database with problem dependent information.
- *Graphical User Interface.* Standardized interfaces for accessing other framework components. Constructed around the shareware package Tk.[20]

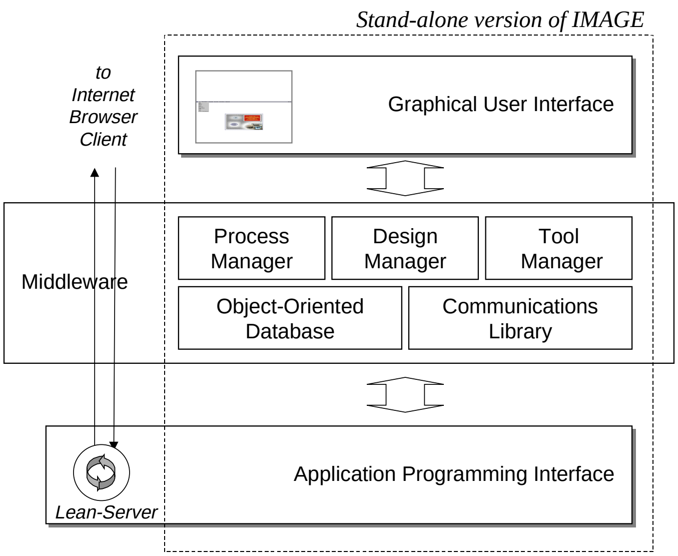

Figure 7. Embodiment of Lean-Server in IMAGE Design Framework

A lean-server has been integrated directly into the API of the IMAGE design framework as shown in Figure 7. The lean-server permits all of the components except for the graphical user interface to be accessed by Web clients. Recall that the web promotes a logical separation of user interface and content. When a lean-server is used, macros are also developed to replace the graphical user interface components found in the stand-alone version. These macros are exercised by the server in real-time to produce HTML for visualization in a Web client such as Netscape™. The macros can also produce other data formats such as a Microsoft™ Excel file. The steps required to integrate the lean-server and to incorporate macros are documented in Reference 14.

One of the many capabilities of IMAGE is the capability to re-configure design processes through the ordering of resources. This can be done using the graphical user interface in the stand-alone version of IMAGE and is now accessible by Web browsers through a lean-server enabled version. A screenshot depicting the meta-design of an aerodynamic procedure is shown in Figure 8. Modules for the advanced design methods appear on the left, available domain tools (agents) to the right of that, and the as-configured process to the far right. The process can be re-structured through a design structure matrix similar to that found in DeMAID.[21] The interface is fully dynamic and immediately available to many users in distributed geographic locations. This simple example goes beyond many other web demonstrations where CGI scripts are executed through static links on a page. Here the design process is fully configurable.

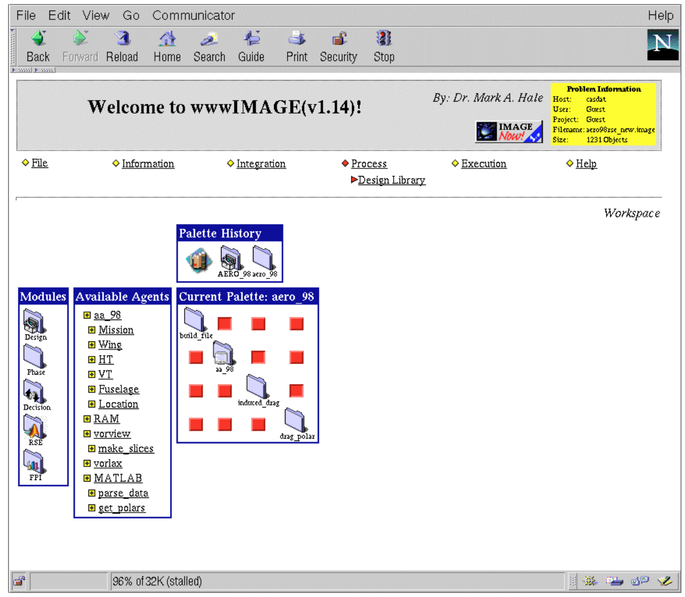

Figure 8. Design Process Configuration Using Lean-Server Enabled Version of IMAGE

ACCESSIBILITY OF ADVANCED DESIGN METHODS

There are a number of emerging design methods to aid designers in finding affordable engineering solutions in the presence of risk and uncertainty. This entails the systematic exploration of stochastic solutions, assessment of the impact of sub-system technologies, and the sizing and synthesis of revolutionary vehicle concepts. Researchers at the Georgia Institute of Technology have been working on devising a toolset that facilitates the design process through the implementation of the new design methods using in-house and extending off-the-shelf software. The tools facilitate approximation, determination of probabilities, optimization, and trade study analysis.

Design frameworks provide the analysis capability required to exercise these emerging methods. The authors are prototyping these methods in the IMAGE environment. The prototyped methods are validated in the graduate design curriculum and have found their way into a Conceptual Aerospace Systems Design and Analysis Toolkit routinely used by the Air Force Research Laboratory for technology assessment.[22]

Macros are being configured through the lean-server enabled version of IMAGE allowing these design methods to be performed using Web browsers. A screenshot of a probabilistic interface which utilizes a Fast Probability Integration tool is shown in Figure 9.[23] A user sets the random variables, calculations (domain analyses), and system response through the process configuration tools as described earlier. Setup parameters for the probabilistic analysis are modified using the fields located to the right in the screenshot. IMAGE is executed on the host UNIX workstation and the Web client is executed on the client PC. The cumulative distribution function that results from the problem execution is shown in Figure 10. A designer has information immediately at her fingertips regarding the likelihood of achieving a function value (depicted by the cumulative distribution function) and the sensitivity of that outcome to the random variables (depicted by the sensitivity bar chart located underneath the graph).

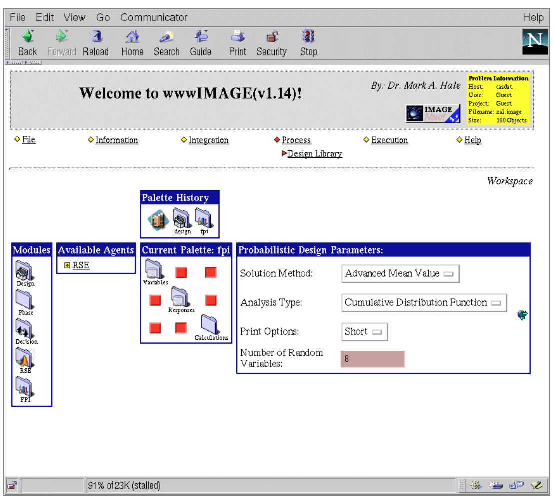

Figure 9. Probabilistic Interface

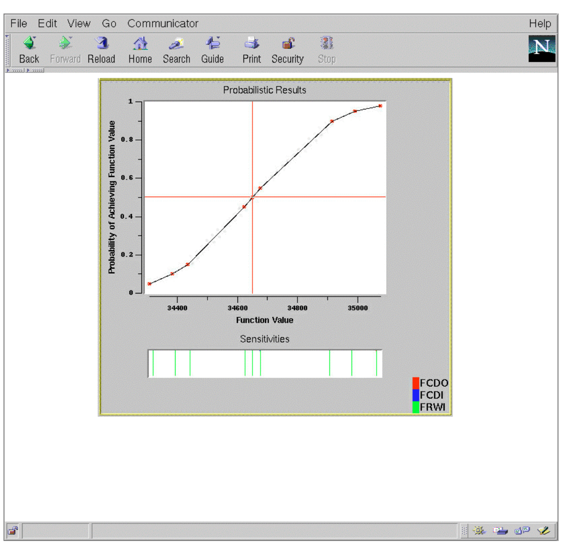

Figure 10. Probabilistic Results for Notional F16

This probabilistic design example shows a straightforward application of the lean-server approach. A designer can easily setup parameters and configure analyses from anywhere on the internet. The preservation of design state within the IMAGE framework is difficult to depict in the screen shots but is necessary functionality overlooked in other internet demonstrations. The importance of design management in modern design frameworks was emphasized in the introductory sections of this paper. This capability permits a designer to return to a design planning activity even though the Web client session is terminated, for instance if the user goes home at the end of the day and closes the Web browser. A designer can also connect from additional sites or open multiple clients, each pointing to a different area in the design system as done in a trade-study analysis. Finally, multiple designers can connect from multiple sites to work within the design system.

APPLICATION TO COLLABORATORIES

Different areas of functionality to be facilitated by design frameworks are described in the introductory sections of this paper. Lean-servers provide a solution for providing design framework services on the internet and thus address the functionality concerns overlooked by other internet demonstrations. It is also shown how advanced design methods can be accessed through this interface and hooks exist in the lean server to promote knowledge management. Through this example, several precursors to the formation of design collaboratories became apparent. Research has been started at the Georgia Tech Aerospace Systems Design Laboratory to address these deficiencies.

Design Portal

The ability to structure and design problems by users in geographically distributed locations is a complicated task indeed. Few, if any, methods permit several users to simultaneously configure a problem for solution. Problems are sufficiently complex that variables, goals, and constraints must be identified and set by users throughout the enterprise. Design portals are being explored as a mechanism for managing group design tasks and may lead to a formalism for collaboration. It is envisioned that portals can guide a user to locate domain dependent views of a problem under study, facilitate the interactions among domains, and serve as a gateway for capturing corporate knowledge.

Collaborative Design Methods

Design methods algorithms are inherently designed for single-user execution until problem completion. For example, if a user configures a gradient-based optimizer to solve a problem and starts the problem executing, the application must typically be run until completion. Few design methods permit users to exit a problem session and either a) have the problem continue executing and enable the user to connect to the problem at a later time or b) have the problem pause until the user returns. (This is not the same as permitting parallel execution or job tracking.) A preliminary study in which a Design of Experiments is performed asynchronously has shown that these scenarios are supported by the lean-server approach.[24]

CONCLUSION

Design frameworks are important for incorporating disciplinary analyses throughout design processes. They are used to manage disparate resources and complex system data. The coupling of advanced design methods within these frameworks was shown to provide powerful decision-making potential.

A historical perspective on framework development is given in this paper in order to illustrate functional achievements that lead to a goal of enterprise-wide collaboration. The need for scalability of analyses to high-fidelity, multi-disciplinary capability drives the technological advances that are needed. The IMAGE framework that is described in this paper demonstrates how a design framework can be equipped with internet services so that distributed users can access the framework using standard Web browsers. This is accomplished through a new lean-server technology developed at Georgia Tech.

Attention needs to be turned to the goal of developing collaboratories – mixed human-machine environments that permit the seamless exchange of experience and rational for improved decision-making. Two shortcomings of existing technologies are highlighted in this paper that hinder collaborative activities. These include configuration-level access by all players to design processes through a design portal and the capability to support intermittent design activities through the use of collaborative design methods.

ACKNOWLEDGMENTS

The authors wish to acknowledge Mr. Dennis Carter of the Air Force Research Laboratory (F33615-95-D-3800), Mr. Samuel Dollyhigh of the NASA Langley Research Center (NAG-1-1786), and Ms. Katherine Drew of the Office of Naval Research (N00014-91-1-0783) for research support for this work.

The authors also wish to thank the students and staff of the Aerospace Systems Design Laboratory for testing the products of the computational research: Andy Baker, Oliver Bandte, Ivan Burdun, Miguel Cartagena, Luis Conejan, Debora Daberkow, Dan DeLaurentis, Adam Frank, Elena Garcia, Brian German, Nate Hines, Peter Hollingsworth, Jesus Jimeno, Jeff Johannsmeyer, Michelle Kirby, Sangjun Ko, Matt Largent, Hahn-Tack Lee, Seung-Jae Lee, Zhimin Liu, George Mantis, Songtao Qiu, Daniella Raveh, Bryce Roth, Julien Scharl, Danielle Soban, Jimmy Tai, Vitali Volovoi, Linda Wang, Neil Weston, Stephanie Willing, Wei Zhao, Scott Zink, and Jason Zumstein.

REFERENCES

1 McCullers, L. A., “FLight OPtimization System, User’s Guide, Version 5.41,” NASA Langley Research Center, December 1993.
2 “Acsynt Overview and Installation Manual,” ACSYNT Institute, Virginia Polytechnic and State Institute, May 1992.
3 Olson, E.D., Mavris, D.N. "Development of Response Surface Equations for High-Speed Civil Transport Takeoff and Landing Noise," World Aviation Congress and Exposition, Anaheim, CA, October 13-16, 1997. SAE-975570.
4 DeLaurentis, D. D., “A Probabilistic Approach to Aircraft Design Emphasizing Stability and Control Uncertainties," Doctoral Thesis, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA. November 1998.
5 Maira, R., “IDAS User’s Manual,” NA-82-467, Volume I, Rockwell International – North American Aircraft, April 30, 1991.
6 Stettner, M., “Tiltrotor Multidisciplinary Optimization,” Doctoral Thesis, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA. August 1995.
7 Jones, K. H., D. P. Randall, and C. K. Cronin, “Information Management for a Large Multidisciplinary Project,” 4th AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, Cleveland, OH, September 21-23, 1993. AIAA-92-4720.
8 Goldin, D. S., S. L. Venneri, and A. K. Noor, “Beyond Incremental Change,” IEEE Computer, pp. 31-39, October 1998.
9 Sanders, P., “Simulation Based Acquisition: An Effective, Affordable Mechanism for Fielding Complex Technologies”, Program Manager, DSMC, Sept.-Oct. 1997.
10 Allen, G.A. Jr., P.J. gage, and E. Venkatapathy, “A Web-based Analysis System for Planetary Entry Vehicle Design,” 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization,. St. Louis MO, September 2-4, 1998. AIAA 98-4826.
11 Acton, D.E. and J.R. Olds, “Computational Frameworks for Collaborative Multidisciplinary Design of Complex Systems,” 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization,. St. Louis MO, September 2-4, 1998. AIAA 98-4942.

12 Becker, J. C., Bloebaum, C. L., "Distributed Computing for Multidisciplinary Design Optimization using Java as a Web Interface", 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellevue, WA, September, 1996.

13 Hulme, K. F., Bloebaum, C. L., "Development of CASCADE: A Multidisciplinary Design Test Simulator", presented at the 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellevue, WA, September, 1996.

14 Hale, M.A., Craig, J.I., and Mavris, D.N., "A Lean-Server Approach to Enabling Collaboration Using Advanced Design Methods", Proceedings of the ASME Design Technical Conferences ’99, Sept. 12-15, 1999, Las Vegas, NV. DETC99/DAC-8597.

15 Proposed for Adoption Fall 1999: HLA Rules IEEE 1516.1, I/F Spec IEEE1516.2, OMT IEEE 1516.3.

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

17 Ousterholt, J.K., An Introducction to Tcl and Tk. Addison-Wesley Publishing Company, Inc., Reading, MA, 1993.

18 “[incr tcl],” Version 1.5, AT & T Laboratories, 1993.

19 Beguelin, A. L., J. J. Dongarra, G. A. Geist, W. C. Jiang, R. J. Mancheck, B. K. Moore and V. S. Sunderam, "Parallel Virtual Machine," Version 3.3, University of Tennessee, Oak Ridge National Laboratory, Emory University, June, 1993.

20 Welch, B. B., Practical Programming in Tcl and Tk, Prentice-Hall PTR, Upper Saddle River, NJ, 1995.

21 Rogers, J. L., "DeMAID - A Design Manager's Aide for Intelligent Decomposition User's Guide," NASA TM 101575, NASA Langley Research Center, March 1989.

22 Hale, M. A., D. N. Mavris, and D. L. Carter, “The Implementation of a Conceptual Aerospace Systems Design and Analysis Toolkit,” 1999 World Aviation Congress and Exposition, San Francisco, CA. SAE Paper Number 1999-01-5639.

23 “FPI User’s and Theoretical Manual,” Southwest Research Institute, San Antonio, TX, 1995.

24 Hale, M. A. and D. N. Mavris, "A Lean-Server Foundation for Collaborative Design," 5th NASA National Symposium on Large-Scale Analysis, Design and Intelligent Synthesis Environments, Williamsburg, VA, October 1999. Submitted to International Journal on Advances in Engineering Software.