AIAA - 98-4825

AEROSPACE TECHNOLOGY ASSESSMENT SYSTEM

Duane E. Veley[*], Maxwell Blair[†] and Jeffrey V. Zweber[‡]

Air Force Research Laboratory, Air Vehicles Directorate,
Structures Division
Wright-Patterson Air Force Base, Ohio 45433-7542

Abstract

The Aerospace Technology Assessment System will be a computer-based environment in which the impact of new technologies on aircraft design, performance and cost can be assessed against baseline configurations. As a minimum, the Aerospace Technology Assessment System will be able to assess a limited set of new technologies at the conceptual and preliminary design levels in the areas of aerodynamics, aircraft layout, structures, controls and cost modeling against a 1995 baseline technology set. The underlying purpose, the architecture of the programming environment and planned capabilities of the Aerospace Technology Assessment System are presented.

Introduction

As is evident from the existence of this symposium, its predecessors and other conferences like it, much work is being done today in the area of integrating functional disciplines in analytical, design and optimization environments. Over the past decade and a half, the change from stand-alone unidisciplinary codes to more highly integrated multidisciplinary codes has been observed (*e.g.*, ASTROS, SPECTRUM, enhancements to NASTRAN, *etc.*) However, it is evident, by the continuing research in this area, that these codes fall short of the multidisciplinary analysis, design and optimization objective. Current technology falls short in two areas: an understanding of the physics of the interaction of the various disciplines and limits in computer technology that hinder the development of multidisciplinary systems. Many DoD, NASA and industrial organizations realize the need for, and benefits of, integrated multidisciplinary analysis (MDA), design (MDD) and optimization (MDO) systems. For example, the industry-government Aero-Structures-Control Interaction (ASCI) team has a requirement for an MDA environment. The authors believe that the Aerospace Technology Assessment System (ATAS) will meet this requirement as well as the technology assessment needs of various other DoD and NASA organizations. Additionally, it is envisioned that ATAS will complement significant research and development performed by other organizations in the areas of MDA, MDD and MDO.

The Air Vehicles Directorate of the Air Force Research Laboratory (AFRL/VA), has a basic need to assess the new technologies in terms of integrated cost and performance measures. The proposed Aerospace Technology Assessment System (ATAS) will meet this need with a computer-based environment. The Fixed Wing Vehicle (FWV) Technology Development Approach (TDA) of the Director, Defense Research and Engineering (DDR&E) has specified three cost and three performance goals of interest for the fighter/attack class of vehicles:

- no increase in unit production cost at T-1, normalized to air vehicle weight;
- no increase in operations and support cost per flight hour, normalized to air vehicle weight;
- no increase in development cost, normalized to air vehicle weight;
- an 8% reduction in air vehicle weight fraction;
- a 10% increase in lift/drag; and
- a 20% increase in controllable angle of attack envelope.

---

[*] Aerospace Engineer

[†] Research Aerospace Engineer, Senior AIAA Member

[‡] Aerospace Engineer, AIAA Member

These metrics may be of interest to the technologist, but the warfighter is interested in a more direct assessment of the benefits of new technologies. Metrics of interest to the warfighter may be range, survivability, payload and reliability. With the Air Force's increased emphasis on access to space, additional technology assessment metrics may need to be developed for this increased emphasis.

The foundational principal behind ATAS is to integrate the various technologies of the Air Vehicles Directorate in a unified environment that will assess the benefits and detriments of the technology and provide direction for any technology application. The integrated environment of ATAS will enable a conceptual level assessment of an aircraft to the extent of, as a minimum, the analytical capabilities of current conceptual design tools (*e.g.*, ACSYNT, CASP, FLOPS, *etc*.) In addition, ATAS will have some preliminary and detail assessment capabilities. New technologies can be modeled in the ATAS environment so that the effect of the technology on a new or existing aircraft may be assessed either alone or in conjunction with other technologies.

Technology assessment in an integrated design environment produces the design guides which can get a new technology on an aircraft. Technology assessment requires trusted and capable design tools. These design tools need to be developed along with the underlying technologies to get the technology on an air vehicle. They will achieve their maximum utility if they are implemented in an integrated design environment. It is envisioned that, in addition to operating as a technology assessment tool, ATAS can form the basis for such an integrated design environment. Many of the methods that will need to be developed for the assessment tool can be used in a production design system.

Complex integrated software systems have been attempted in the past, but due to shortcomings in computer technology, have not come to fruition. Recent developments in computer technology including new computer languages/programming environments will make ATAS a viable system. Those developments that are needed to make such an effort fruitful are given in the section on architecture capabilities.

One important milestone in the development of multidisciplinary analysis and optimization tools is the Automated STRuctural Optimization System (ASTROS) (Neill and Herendeen, 1996; Neill et al, 1996a; and Neill et al, 1996b) developed in the 1980's and early 1990's by the Air Force. From the ASTROS effort several lessons are learned. First, there are great advantages to having the different disciplines sharing a common database. Second, the language in which the backbone of the system, or the system driver, is written (in the case of ASTROS this would be MAPOL) should be one in which computations can be performed. That is to say, the backbone simply does not link various codes together, but may also perform some of the common types of analyses. Third, optimization can be implemented as a module like any other discipline.

Even though these lessons are positive and show ASTROS making significant improvements over the mid '80's state-of-the-art, recent developments in computer science allow improvements in the ways that these features can be implemented. First, object-oriented databases allow greater flexibility, and easier access to various types of engineering data. Second, demand-driven calculations (lazy evaluation) when combined with object-oriented modeling allow much easier addition of new analysis capabilities. This is opposed to sequential programming in which modifications to the code (*e.g.*, adding new modules) requires modifying the sequence, recompiling the sequence and relinking the program. Modification of the sequence is not always straight forward and may require breaking the technology into several pieces and integrating each piece in more than one location so that the sequential flow of data is appropriate.

Object-oriented programming has opened a new door to developing multidisciplinary design tools. Object-oriented programming helps the programmer to organize the information that is required to do an analysis or design. For example, an object such as an airplane would include subobjects such as wing, tail and fuselage. The wing object in turn could contain subobjects such as ribs, spars and control surfaces and properties such as span and chord length. But the object need not be limited to properties of one discipline. A wing object could also contain aerodynamic, structural and control properties and subobjects that include subsystems in addition to the other disciplines. Thus, object-oriented programming not only helps organize information, it also helps integrate the information. When combined with features such as dependency tracking and demand-driven calculations, object-oriented programming provides a superior environment.

Another development that has made significant impact on multidisciplinary optimization is the ability to interface legacy codes more readily. TCL/TK, a scripting language, is probably the best known language for such applications. Recently several engineering design environments (*e.g.*, iSight, Image) have been built on TCL/TK as a way of integrating computer programs to perform multidisciplinary analysis. Other recent engineering design environments (*e.g.*, AML, SIDE, DIS, ICAD and Modelogic) use other technologies to integrate legacy codes into a multidisciplinary analysis/design/optimization environment.

Finally, it must be noted that system level assessments of some technologies may require detailed, high-fidelity models and simulations of the phenomena. Accordingly, ATAS should facilitate the feedforward and feedback of design information between various levels of modeling.

Capabilities of ATAS — Baseline Technologies

Many disciplines are required for the analysis and design of a baseline air vehicle and, depending on the specific technology, many of those same disciplines are required to assess the impact of new technologies on the air vehicle. Failure to include some disciplines in the assessment process may cause some synergistic or detrimental effects to be overlooked. This section lists some of the disciplines that are needed, specifically the ones that are of most interest to the Air Vehicle Directorate for the assessment of an air vehicle with baseline technologies.

- Aerodynamic Analysis – Analysis techniques suitable for interior and exterior flow should be incorporated into ATAS. These techniques should be capable of analyzing the entire vehicle and major components alone. Techniques should be provided for the analysis of both steady and unsteady flows. Techniques should be provided for boundary layer and drag calculations. The following four classes of aerodynamic analysis techniques should be provided:
    - Linear Potential Flow.
    - Transonic Small Disturbance.
    - Euler.
    - Navier-Stokes.
- Structural Analysis – Analysis techniques for determining stresses, strains, displacements, natural frequencies and mode shapes should be incorporated into ATAS. These techniques should be capable of analyzing isotropic and orthotropic materials. These techniques should be capable of determining the responses to static and time varying loads and boundary conditions. The following three classes of structural analysis techniques should be provided:
    - Equivalent Beam and Equivalent Plate (*e.g.*, 1st order analysis of 1 and 2 dimensional models).
    - 1-D FEM (Rod/Bar Elements).
    - 2-D FEM (Plate/Shell Elements).
    - 3-D FEM (Solid Elements).
- Aeroelastic Analysis – Analysis techniques for coupled fluid-structures problems should be incorporated into ATAS. These techniques should use the aerodynamic and structural analysis techniques above to solve those portions of the coupled problem. Solution techniques for the following two classes of aeroelastic problems should be provided:
    - Static response – Divergence, flexible trim, flexible control power and equilibrium steady flow.
    - Dynamic response – Flutter and arbitrary time-dependent flow.
- Control System Development – Analysis techniques for coupled fluid-structures-controls (*i.e.*, aeroservoelastic) problems should be incorporated into ATAS. These techniques should use the analysis techniques from the other technology disciplines that are needed. Solution techniques for the following three classes of problems should be provided:
    - Stability derivatives – Rigid and flexible.
    - Flying/handling qualities analysis – This should cover sizing control surfaces, actuators, sensors and other control system components.
    - General vehicle dynamics (*i.e.*, performance and stability and control) for arbitrary maneuvers.
- Electromagnetic Analysis – Analysis techniques for determining the infrared and radar signature of the entire vehicle and major components alone should be incorporated into ATAS.
- Power Management Analysis – Analysis techniques for determining the hydraulic and electrical power requirements of the entire vehicle and major components should be incorporated into ATAS.
- Cost Modeling – Both activity-based and parametric cost modeling techniques should be incorporated into ATAS. ATAS should be capable of generating various cost reports and breakouts of the cost data from models created with each technique or a hybrid model created using both techniques together. In addition to reporting the FWV TDA cost goals, ATAS should report costs indicative of the following list of candidate cost breakouts:
    - Tooling cost.
    - Labor cost.
    - Material cost.
    - Indirect cost.
    - Airframe structure cost.

    - Subsystems cost.
    - Engine cost.
- Weight Modeling – Both parametric and bottoms-up weight modeling techniques should be incorporated into ATAS. ATAS should be capable of generating various weight reports and breakouts of the weight data from models created with each technique or a hybrid model created using both techniques together. In addition, ATAS should be capable of calculating center of gravity and moment of inertia. The following is a list of candidate weight breakouts:
    - Airframe structure weight.
    - Payload weight.
    - Fuel weight.
    - Avionics weight.
    - Propulsion system weight.
- General Framework Tools – The following general capabilities should be incorporated into ATAS.
    - Sensitivity Analysis – The automatic differentiation of any equation that is written in the underlying architecture of ATAS. Finite difference of any analysis result that requires a solution by a code external to the underlying architecture of ATAS.
    - Optimization – Gradient-based methods, non-gradient-based methods (*e.g.*, genetic algorithms, simulated annealing), hybrid methods (*e.g.*, topological optimization) and multiobjective function methods.
    - Statistical Techniques – Methods for generating response surfaces (*e.g.*, design of experiments) and tabular datasets (*e.g.*, drag polars)
    - Plotting and Visualization – Carpet plots, fringe plots and line plots.
- Known omissions:
    - Survivability/Damage Tolerance.
    - Avionics/C^4I.
    - Propulsion.

While these technologies are essential for developing a baseline air vehicle technology assessment tool, they are by no means exhaustive as is indicated by the last category of known omissions.

The development/integration of new disciplines in ATAS will be driven by the selection of a configuration and mission against which to assess the technologies as well as the suite of technologies chosen to be assessed.

Capabilities of ATAS — Functional

Functional capabilities are those that integrate the technology disciplines as needed for analyzing/designing/optimizing aircraft. ATAS needs to have a minimum functional capability in order to be able to do a good job at assessing technologies in a user-friendly, integrated environment. This minimum functional capability of ATAS is preliminarily slated to include:

- Vehicle Layout – An interactive system for laying out the vehicle should be incorporated into ATAS. This system should build and/or modify the object hierarchy and parametric relationships that represent the vehicle.
    - Conceptual-Level – The system should have a suite of preexisting parametric objects (*e.g.*, engine, wing, tail, landing gear, fuel volume, payload volume) that are capable of representing an entire conceptual-level model of modern fighter/attack and bomber/airlift/patrol class air vehicles. In addition, the system should have CAD-like parametric geometric primitives that can be used to build new objects needed for a conceptual-level model of revolutionary air vehicles (*e.g.*, uninhabited air vehicles, transatmospheric vehicles, hypersonic air vehicles).
    - Preliminary-Level – The system should be capable of generating a vehicle model that is suitable for preliminary-level analyses in the technologies incorporated in ATAS. Some examples are:
        - Aerodynamic Model – A watertight model should be generated for internal and external flow analysis of the entire vehicle and major components.
        - Structural Model – Major substructural components (*e.g.*, ribs, spars, bulkheads, stringers) should be placed. These substructure models should be, at a minimum, suitable for a 2-D FEM analysis of the entire vehicle or major components.
        - Control System Model – Models of moveable control surfaces should be generated. These models should be incorporated into the preliminary-level aerodynamic and structural analysis models.
        - Subsystems – Models of hydraulic, electric and fuel subsystems should be generated. These models should include, at a minimum, routing of hoses (fuel and

    hydraulic), routing of wire harnesses, placement of pumps, placement of generators, and placement of major consumers of hydraulic or electrical power.

        - Electromagnetic Model – Models should be generated for infrared and radar analysis of the entire vehicle and major components.

- Perform Mission Range Analysis – A mission analysis module should be incorporated into ATAS. This module should determine either the amount of fuel burned for a given mission profile, drag polar, engine deck and weight breakout or the maximum length of certain mission segments (*i.e.*, range) for a given fuel quantity, drag polar, engine deck and weight breakout. The module should only specify the interface requirements for the models of the drag polar, engine deck and weight breakout.
    - Layout Mission Profile – The mission profile should be input as segments.
        - Takeoff – Should have options to compute takeoff ground roll distance and distance to clear 50 foot obstacle for best and fixed power settings.
        - Climb – Should have options to climb for minimum time, minimum fuel, minimum range and climb at a constant speed. Should have ability to climb to best altitude for cruise.
        - Accelerate – Change speeds at a constant altitude. Should have ability to change speed to best speed for cruise.
        - Cruise – Should have options for fixed speed and altitude, best altitude for a fixed speed, best speed for a fixed altitude and optimum speed and altitude.
        - Loiter – Should have the same options as cruise.
        - Combat – Should have options for fixed time, fixed fuel used and fixed number of turns performed.
        - Land – Should calculate landing ground roll distance and distance from 50 foot obstacles and best and fixed power settings.
        - Warm-up/Taxi – Fuel used at a given power setting for a specified time.
        - Load – Should allow for payload weight change, either with or without engines running.
    - Generate Drag Polar – A dataset should be used that includes the relationships among total vehicle drag coefficients total vehicle lift coefficients and other relevant parameters (*e.g.*, Mach Number, Reynolds Number, *etc*.) These datasets should be consistent with ones that can be produced by the aerodynamic analyses.
    - Download Engine Deck – A dataset should be used that includes the relationships among fuel consumption, thrust and other relevant parameters (*e.g.*, Altitude, Mach Number, *etc*.) Standard dataset formats should be used to the maximum extent practical.
    - Breakout Weight – The weight breakout of the vehicle should be input in three groups; fixed weight, payload weight and fuel weight. These weight breakouts should be consistent with ones that can be produced by the models from the ATAS technical capabilities in weight modeling.
        - Fixed Weight – Weight that must be maintained throughout the mission. This weight should be further broken out to account for major categories of fixed weight (*e.g.*, air vehicle structural weight, engine/propulsion system weight, mission equipment weight, *etc.*)
        - Payload Weight – Weight that can be changed at fixed points in the mission. For example, firing a missile during combat.
        - Fuel Weight – Self explanatory.
- Encounter (Combat) Scenario – The aircraft performance can be assessed through a simulated encounter with other aircraft or missiles. For more information on this subject see Blair (1998).
- Incomplete Model Analysis – ATAS should automatically generate the parts of a model hierarchy that are needed for a desired analysis if they have not already been specified by the user. The system should indicate which portions of the model hierarchy were automatically generated.

Capabilities of ATAS — Operational

Operational capabilities define how the user will interface with the technical disciplines and the system functional capabilites. The operational capabilities should include

- Graphical User Interface – The system should use a graphical user interface (GUI) through which data can be entered. While the ATAS will automate a great deal of the analysis/design processes, human intervention is still required to monitor progress and direct additional analysis requirements. Therefore, the GUI is to aid the human in this process by providing the available options at any step along the way.

- Process initiation – The process of assessing technology must begin at some point. This starting point depends on the aircraft on which the technology is being assessed.
    - Existing aircraft – For aircraft that is already existing, the user must be able to start with the existing configuration. The user should be able to enter an existing design quickly and easily into ATAS.
    - New (notional) aircraft – Technology can be integrated into the aircraft design from the initial concept of the aircraft.
    - Existing ATAS aircraft – A technology may be assessed against a baseline configuration of an aircraft already existing in the ATAS environment (a saved model). This baseline may be selected as the starting point for assess that technology.
- Saving/versioning – ATAS should be able to save the current model of the aircraft at any point in the process and bring that model up again at some later time without having to go through the analysis/design process again. This capability should allow date stamping and version numbering.

- Allowable modifications – This may also be called modification privileges. As a model is distributed amongst several engineers to analyze/design various parts of the aircraft, control over which properties each engineer is allowed to modify must be made. This concept of restricting variation of parameters may also apply to assessing new technologies on existing aircraft where some parameters are not to be changed.

- All aircraft types – Although it is not expected that the first version of ATAS will be applicable for all types of aircraft (*e.g.*, fighter, bomber, transport, UAV, hypersonic, *etc.*), it should be developed in such a way as to permit additional types of aircraft to be assessed with relatively minor modifications. Some technologies that may find their way on to a UAV, may not be cost effective for a transport or bomber.

- Expandability – ATAS must be expandable to include future technology and technologies outside the Air Vehicles Directorate. Not only does this make ATAS a more powerful technology assessment tool, but it also opens the door for ATAS to be the foundation to a commercial design tool.

Capabilities of ATAS — Architecture

Architecture capabilities are the hardware and software computer system capabilities needed to be able to perform all the capabilities of ATAS presented above. As the term architecture is used in this paper, it is meant the architecture of the programming environment in which ATAS is developed and not the architecture of ATAS itself. The architecture provides the foundation for ATAS and adherence to these capabilities will ensure a successful technology evaluation tool and provide a solid foundation for a design tool.

- Commercial Software – The architecture should be commercially supported. It should have a customer base in addition to the government and the software developer. The purpose of this requirement is to reduce the potential of selecting an architecture that will soon fade away.

- Object-Oriented – The architecture should employ an object-oriented programming structure. The purpose and advantage of object-oriented architectures are in its ability to act as language and database, organize the program and to aid in the integration of the different technologies in a logical fashion.

- Support for Multiple Computer Operating Systems – The architecture should run natively on multiple computer operating systems. At a minimum, the architecture should run on SGI IRIX and Microsoft Windows NT. ATAS is not to be tied to a particular hardware platform.

- Adaptive Class Structure – The architecture should allow for the modification of the object hierarchy (including adding, deleting and modifying properties and sub-objects) at any time. The architecture should not require the use of a “superclass”. It is envisioned that during the analysis/design process that the hierarchy of the aircraft will change. Components not initially considered may appear and others may disappear. Also, it is possible that a component/technology that was not a part of ATAS at the beginning of the assessment/design process will need to be incorporated into ATAS.

- Open Architecture – The architecture should be capable of supporting multiple analysis techniques and information standards for any discipline. The government realizes the current investment in CAD/CAM tools and physics-based solvers (*e.g.*, finite element modeling packages, computational fluid dynamics packages, computational electromagnetics solvers, *etc*.). The ATAS architecture should not require a priori the use of any analysis technique or information standard. Additionally, different modules within the architecture should be easily replaceable/maintainable.

- Unified Part Model – The architecture should be capable of modeling the information for multiple disciplines in a single object hierarchy. These objects should take advantage of the adaptive class structure and open architecture to allow real-time addition and modification of the design information.

- Dependency Tracking – The architecture should automatically track the dependencies between various objects and properties within the model. Resultant properties and objects should be "notified" when the properties or objects on which they depend are changed.

- Demand Driven Calculation (*a.k.a.*, Lazy Evaluation) – The architecture should only perform the calculations that are required to determine the result of a desired analysis or function evaluation. Even though an object may contain many properties, the object is not a module. The module for each property is defined by its dependency tracking. Thus, if the user chooses to determine the displacement of the wing under a certain static load condition, then only those properties that are required to calculate that displacement such as the load and the wing stiffness (whether they are properties of the wing or other objects) are calculated and things that do not effect the wing displacement, like stress and dynamic characteristics, are not computed at that time.

- Common Syntax – The architecture should use a common syntax for all disciplines within the architecture. For example, the programming environment should have the same syntax to write graphical user interface applications as it does to interface with foreign analysis routines as it does for basic object-oriented class definition. This requirement should not imply that all external analysis routines will be written in the same language. It does require that the syntax for calling the external routines from the architecture should be consistent. In other words, the syntax should be independent of the underlying foreign application.

- Multiple Simultaneous Users – The architecture should be capable of having multiple simultaneous users interact with a single model hierarchy. The structures, aerodynamics and control specialists should all be able to access the model that is being developed by the conceptual designer at the same time and work on their portion of the model or extract the portion of the model that they need to work in their own environment.

- Network Distributed Model – The architecture should be capable of supporting a single model hierarchy that is distributed across various machines that are connected via a network. The various machines should not be restricted to the same operating system. After extracting a portion of the model from the global model, a specialist can work with that portion of the model on his own machine in the ATAS environment and feed the results back to the global model once his analysis is complete. This is useful for time consuming analyses and/or those processes that require a lot of human intervention. While this may appear to be an over the wall type process, ATAS should maintain the dependencies and the integrity of the entire model to permit an automated communication level between those systems. This feature also helps distribute the database across several machines thus reducing the memory/disk space requirements of any one machine.

- Geometry – The architecture should be capable of displaying the geometry of the objects either inherently or through a geometric modeler closely coupled to the architecture. Geometry plays an important role in assessing a design. A simple "does it look right" assessment can be quite beneficial. Geometry also helps in interference checks and other assessments that may help answer the question, "Can we build this?"

The presence of these requirements in the software architecture will significantly reduce the risk of developing a technology assessment system. The most critical of these requirements (the enabling requirement) is the adaptive class structure. This requirement implies an object-oriented language and the orderliness that comes about from such a language. But more importantly, the adaptive class structure permits objects to be modified after they have been instantiated, not just the value of properties, but the property formulas and subobjects may be added or removed. This permits low-order approximations to be defined initially, and then as information is gained, a higher fidelity formula may be used.

These architecture requirements were developed to allow for a minimum disruption to the current way of doing business. The open architecture ensures that the investment in legacy tools will not be wasted. Also, engineers will not be forced to learn new analysis packages.

The adaptive class structure and distributed modeling capability fit well with the current air vehicle design process. They allow detail to be added as it is available and they allow multiple engineers to simultaneously work on different parts of the same design.

Summary

Though still in the process of developing requirements for ATAS, this document presents the current thoughts of what is needed to develop an Aerospace Technology Assessment System. These capabilities address issues such as the technology disciplines, functional use of the disciplines, operational interface and the computer architecture. These are placed in a context of a system that can be used to assess the impact of new technologies developed in the Air Vehicles Directorate on air vehicles. Improvements in both computer hardware and software architectures permit programmers to overcome problems of past complex integration programs.

References

Blair, M. (1998), “Enabling Conceptual Design in a Technology-Driven Environment,” Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA Paper 98-4741, St. Louis, MO, Sept. 2-4, 1998.

Neill, D. J. and Herendeen, D. L. (1996), ASTROS Enhancements: Volume 1 – ASTROS User’s Manual,WL-TR-96-3004.

Neill, D. J., Herendeen, D. L. and Hoesly, R. L. (1996a), ASTROS Enhancements: Volume 2– ASTROS Programmer’s Manual,WL-TR-96-3005.

Neill, D. J., Herendeen, D. L. Venkayya, V. B. (1996b), ASTROS Enhancements: Volume 3– ASTROS Theoretical Manual, WL-TR-96-3006.