SAWE Paper No. 3201
Category No. 23

Structural Weight Estimation for Launch Vehicles

Jeff Cerro and Zoran Martinovic
NASA Langley Research Center, Hampton, VA

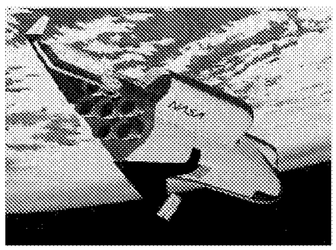
Philip Su
Raytheon Technical Services Co., Hampton,VA

Lloyd Eldred
Swales Aerospace, Hampton, VA

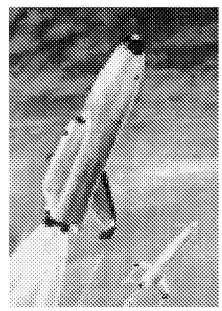
Presented at the 61st International Conference of Society of Allied Weight Engineers, Inc.
Virginia Beach, VA, May 18-22, 2002

Introduction:

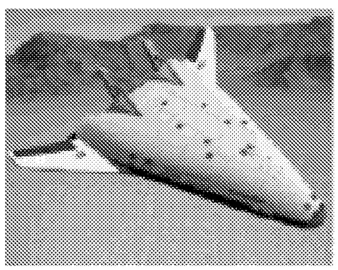
This paper describes some of the work in progress to develop automated structural weight estimation procedures within the Vehicle Analysis Branch (VAB) of the NASA Langley Research Center. One task of the VAB is to perform system studies at the conceptual and early preliminary design stages on launch vehicles and in-space transportation systems. Some examples of these studies for Earth to Orbit (ETO) systems are the Future Space Transportation System [1], Orbit On Demand Vehicle [2], Venture Star [3], and the Personnel Rescue Vehicle[4]. Figure [1] shows some of the concepts the different vehicle types encountered in these studies.



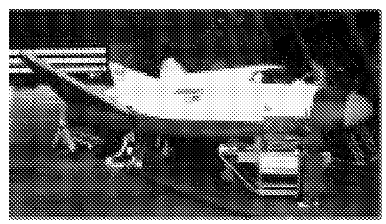
FSTS



Orbit On Demand



Venture Star SSTO



Personnel Rescue Vehicle

Figure 1) Typical VAB Launch Vehicle System Studies

Structural weight calculation for launch vehicle studies can exist on several levels of fidelity. Typically historically based weight equations are used in a vehicle sizing program. Many of the studies in the vehicle analysis branch have been enhanced in terms of structural weight fraction prediction by utilizing some level of off-line structural analysis to incorporate material property, load intensity, and configuration effects which may not be captured by the historical weight equations. Modification of Mass Estimating Relationships (MER's) to assess design and technology impacts on vehicle performance are necessary to prioritize design and technology development decisions. Modern CAD/CAE software, ever increasing computational power and platform independent computer programming languages such as JAVA provide new means to create greater depth of analysis tools which can be included into the conceptual design phase of launch vehicle development. Commercial framework computing environments provide easy to program techniques which coordinate and implement the flow of data in a distributed heterogeneous computing environment. It is the intent of this paper to present a process in development at NASA LaRC for enhanced structural weight estimation using this state of the art computational power.

The Space Launch Initiative:

Currently NASA is in the process of defining technologies required and vehicle configurations necessary to support the Space Launch Initiative (SLI) [5]. The Space Launch Initiative is also referred to as the 2nd Gen. Program in that its goal is to define the technologies and possible vehicle architectures for a system of launch and upper stage vehicles which will replace our first generation reusable launch system, the Space Shuttle. A major task of SLI is to study vehicle concepts and the technologies required to enable these concepts with the overall goal of high safety standards and low life cycle costs. Goals of a launch capability with a 1 in 10,000 loss of crew level of reliability, and an operational cost of $1000.00 per pound of payload to orbit have been established. It is desired to develop technologies for such a system to the point where vehicle design can begin by approximately the year 2005. To help in this assessment and enable NASA to be an informed consumer of industry proposed commercially based launch systems, the agency wishes to perform internal assessments of its own and of contractor defined vehicle architectures. No single vehicle will likely be able to satisfy the SLI defined mission requirements, so a set of vehicles comprised typically of launch vehicle stages, entry vehicles, and orbital elements are required to work together. A suite of vehicles which can meet mission requirements is being called a launch vehicle architecture. In the area of weights and vehicle sizing it is desired to quantitatively assess the impacts of structures technologies on architecture element performance. Structural mass estimation techniques which can account for differing vehicle structural arrangements, structural concepts, material choices, loading and failure mode/factor of safety inputs are required. This data can not be totally derived from historical data as the proposed concepts will differ too much in terms of vehicle architecture, structural arrangement, and structural concepts to have direct correlation to historical weights. To uniformly assess a diverse array of in-house and contractor proposed vehicles at the same level of fidelity is the goal of such a structural mass estimating analysis. Such a system, when properly calibrated, will provide designers with quantitative techniques which can be used to modify system design variables and obtain optimum vehicle configurations. It will also provide the government some insight into the assumptions and fidelity of contractor supplied vehicle weight statements. These repetitive and complex analyses only have a chance of being performed efficiently and fairly if automatic procedures can be put in place to implement the required analysis processes.

Program support for Tool Development:

Recognizing the importance of making correct decisions early in the architecture definition process, NASA has programs in place to support the development of conceptual and preliminary design systems for 2nd Gen RLV's. NASA Langley's High Performance Communications and Computing Program (HPCCP) [6] was used in FY2001 to automate analysis procedures and provide a framework for large scale vehicle design data processing. The program included development of techniques for three main processes, the first was vehicle sizing coupled with layout, aerodynamic and trajectory analyses. Second was the automated creation of an aerothermodynamic environment for the entry problem and using that heating environment to size vehicle thermal protection systems (TPS). A third element of the program was an implementation of finite element analysis (FEA) and automated Computer Aided Design (CAD) procedures to calculate structural mass properties. The HPCCP program was ended in 2001 but some elements of it, including some of the structural mass estimation procedure development are being continued. A current program, the Advanced Engineering Environment (AEE) is in place to focus tool development efforts across the NASA centers and bring well defined processes into a Product Data Management (PDM) controlled computing framework. AEE structural weight estimation may have several levels of fidelity. Bottoms up weight prediction based on beam-theory techniques is one level of analysis where rapid turnaround should be easily obtained. Three dimensional finite element modeling, internal loads assessment and structural element sizing is a higher level of structural simulation which should also eventually fit into the AEE toolbox. The FEA based process in it's goals is similar to the Boeing Corporations RAMPAGE program [7] but at a somewhat reduced level of modeling fidelity. It is the intent here to first develop a system which can be exercised in the AEE environment, yet maintain sufficient generality and modeling flexibility such that the growth path to multiple vehicle configurations, and higher fidelity modeling are supported. Eventually these "conceptual" procedures may provide the starting point for preliminary design as much of the same data structures (i.e.: FEA data) will be utilized in both design phases. A structural weight estimation system with these characteristics which implements modern computer science programming capabilities in a heterogeneous collaborative design environment will be maintainable and useable for years beyond the 2005 timeframe.

Structural Considerations:

Bottoms up weight estimation for launch vehicle structural systems requires the definition of four basic design decisions to provide enough detail for discriminating calculations to be made. First a vehicle structural arrangement must be defined. Also referred to as a "big-bones" layout or structural skeleton the structural arrangement is a description of the components and their interconnectivity which define major structural load paths. For a reusable launch vehicle this would mean the components needed to efficiently transmit propulsion and lift forces from the body and wing into the required subsystem masses. For a launch vehicle with cryogenic tanks basic decisions have to be made regarding their incorporation into the structural arrangement because tanks comprise a large volumetric requirement of the vehicle. A cryogenic tank can be either integral or non-integral with regard to supporting overall vehicle fuselage body bending loads. Tanks which are supported in a statically determinate manner within an encompassing fuselage structure are termed non-integral. Non-Integral tanks can be desirable in that they eliminate tank thermal growth influence on vehicle shape and internal force generation. There are of course, many trades in the area of structural arrangement. Large non-integral tanks also imply large regions of confined space between tank and fuselage structures which are operational hazards and although clean in terms of structural loadpath are degrading in another sense to vehicle safety and reliability [8]. An integral tank concept is one in which the large cryogenic tanks carry overall fuselage bending loads, as well as internal tank ullage and head pressure loads. This was the design of the X-33 vehicle. Note that even though the X-33 utilized an integral tank structural arrangement, because of cross section differences between the multi-lobe cryogenic tanks and the flat aerodynamically shaped OML there were still confined spaces created by the offset thermal protection system. Other structural arrangement decisions might include:

- A choice between LOX tank forward or aft of the LH2 tank, this is a trade off in ascent controllability (better if LOX forward) vs. LH2 tank weight (better if LOX aft)

- Consideration of how the wing structure is attached to the fuselage, full carry through or utilization of tank ring frames to carry wing root bending moments. Largely a tradeoff in aerodynamic performance vs. structural weight. Is it better if the tank frames can carry wing bending and so a minimal amount of fairing drag is encountered or is a structurally more efficient system using carry through structure but with the wing below the cryogenic tanks.

- A choice between central vertical fins or wing tip-fins.

- A choice between fuselage integrated payload regions, such as in the STS orbiter, or having an externally attached payload support structure.

The procedures to be described in this paper will have the ability to handle all of the structural arrangement trades listed above.

The second basic design decision is structural concept, that is a definition of the wall construction method used for each component enumerated in the structural arrangement. As an example of structural concept definition a design might consist of a composite semi-monocoque nose structure, aluminum monocoque LOX tank, aluminum-lithium isogrid intertank adapter, integral skin-stringer aluminum-lithium LH2 tank, and metal-matrix frame-stringer stiffened thrust cone. Trade-offs in structural concept might also compare a structure made of conventional metal manufacturing techniques to a structure made with state of the art low part count composite processing techniques. Decisions on manufacturing technology priorities based upon their impact on vehicle performance can be made if performance sensitivity to those manufacturing techniques are quantifiable. The wing too will have structural concept choices, typically separate choices for upper surface skins, lower surface skins, ribs, and spars.

Material property specification itself is the third basic element requiring definition in the conceptual design structural weights calculation process. Provision must exist in the design system to implement tables of material property data. With this flexibility in place the performance gain of advanced material systems can be quantitatively assessed.

With the vehicle structural arrangement, structural concepts, and material properties defined the remaining data needed to be able to perform structural analysis based weight estimations are vehicle loads and design criteria. The proposed system will have quantifiable sensitivity to design load conditions and thus vehicle load factors. Propellant tank pressure stabilization assumptions could also be assessed. In general, loads, factors of safety, and failure mode considerations will have quantifiable affects on structural mass estimation. Table I lists load cases typically considered at the conceptual level of bottoms up structural weight assessment.

Table I – Design Load Cases

1) Proof Pressure
2) Ten Day +Z Wind on Pad, unfueled condition
3) Ten Day -Z Wind on Pad, unfueled condition
4) One Day +Z Wind on Pad, fueled condition
5) One Day –Z Wind on Pad, fueled condition
6) Liftoff
7) Max Dynamic Pressure
8) Max Wing Normal Force
9) Max Axial Acceleration
10) Subsonic Entry Maneuver
11) Main Gear Touchdown
12) All Gear Touchdown
13) Ground Handling in the Horizontal Condition, unfueled, unpressurized

Technical Approach, Overview:

Based on the requirement of having the capability to quantitatively assess the above four areas of structural definition, and upon previous experience in the VAB with finite element analysis application to structural weight prediction, a plan was developed which would integrate FEA analysis into the tool set used for conceptual vehicle design.

It is recognized that good weight estimation results can be obtained through the use of beam type models [9] [10] and indeed as previously mentioned such processes are being developed for the AEE environment. However to set the stage for more advanced analyses and prepare utility programs and procedures which can be migrated to the preliminary design level, general 3D shell element based finite element analysis tools need to be developed. Using a two dimensional shell element based system which models in three dimensional space forces the development of computer programs and data structures to work with finite element data structures more suitable to higher level analysis. Initially the models have been very simple and therefore relatively easy to automate for conceptual design level modeling. Future expansion to more complex general 3D modeling should include integration of beam and solid element types. Migration to this level of fidelity will be made easier by starting out with the types of file structures and general data manipulation required by the shell element modeling technique. Such model fidelity also paves the way for upgrades whereby major structural cutouts, frames, bulkheads and complex loading conditions can be easily handled. There will be a well defined growth path to go from conceptual level vehicle design to early preliminary and possibly provide links into the detail design phase of vehicle manufacturing.

Utilizing commercial CAD/CAE software such as SDRC IDEAS [11] also provides well defined data structures, file formats and procedures that incorporate flexibility and growth potential into the design process. For example the ability to define multiple load cases, define multiple boundary conditions, keep files of material properties, and implement varying structural design concepts over regions of the vehicle are procedures routinely handled in a commercial FEA program. In the future the data structures will have been in place which will be useful to implement thermal loading as the structures process becomes more integrated with vehicle thermal analysis processes.

Modeling an entire vehicle as shell elements also enables creation of a data file of running loads for each design load case which varies over all of the structural components that make up the vehicle. Structural sizing of elements defined in a commercial data file format such as an IDEAS universal file is performed using the HyperSizer [12] structural design software. This element sizing program takes industry standard FEA data files including internal element loads and determines the required structural sizes to support multiple structural load conditions. Detailed strength and stability failure mode checks for specified wall construction techniques such as monocoque, and numerous semi-monocoque construction techniques are performed by the program. Calculated element structural weight, assembly level weight summations, failure mode, dominant loadcase and other such design information is accessible in the HyperSizer program. Structural weights for components calculated by HyperSizer need to be factored to account for features not inherent in the analysis such as joints and other non-optimum structure. Calibration of this transition from theoretical to as-built structural weights for various launch vehicle components is required.

The entire process of gathering input information, creating a representative finite element model, defining structural arrangement, structural concept, material properties, load conditions, performing element sizing and making new estimates of vehicle structural weight has been implemented in the commercial software framework environment called Model Center [13]. Model Center provides software tools to link computer code on heterogeneous server computers. Client computers display a GUI representation of a user defined process flow and permit modification to the process if required. This GUI also provides a means to link variables between computer codes on the server machines.

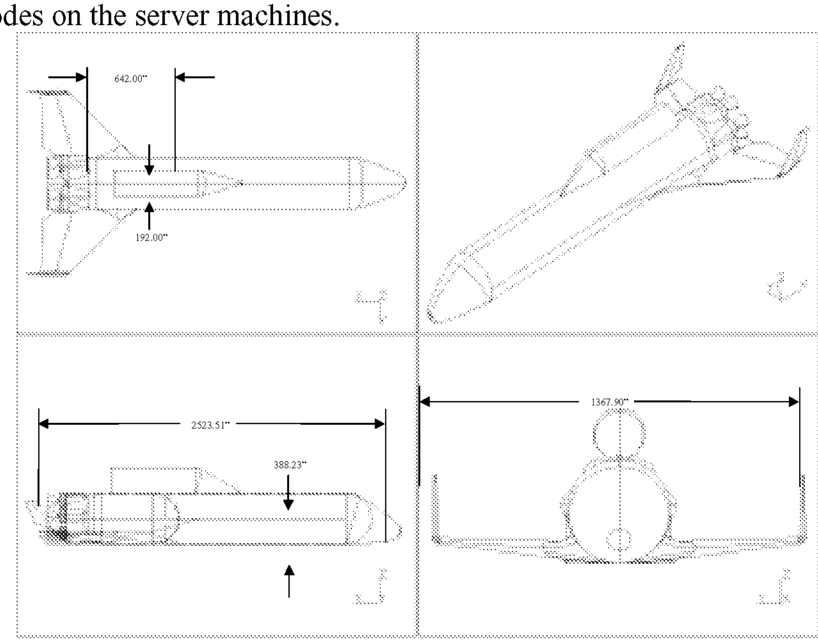


Figure 2) – Layout of Example Vehicle, SSTO with an external payload attached

Technical Approach, Implementation:

In this paper we have chosen to implement the automated weight estimation process on the example vehicle configuration shown in figure 2. This is a Single Stage to Orbit vehicle (SSTO) consisting of a basically cylindrical fuselage, aft mounted low aspect ratio delta wing, and vertical tip-fin controllers. Internally an integral liquid oxygen (LOX) tank is positioned behind an integral liquid hydrogen (LH2) tank. The payload is attached as a separate structure on back of the integral cryogenic tank fuselage structural arrangement.

Simplistically this vehicle can be described by the 7 major structural components shown in figure 3. Note that the payload support structure is not yet one of the generic structural components in the system and lumped masses are used to create payload and payload pod inertial loads in the fuselage structure.

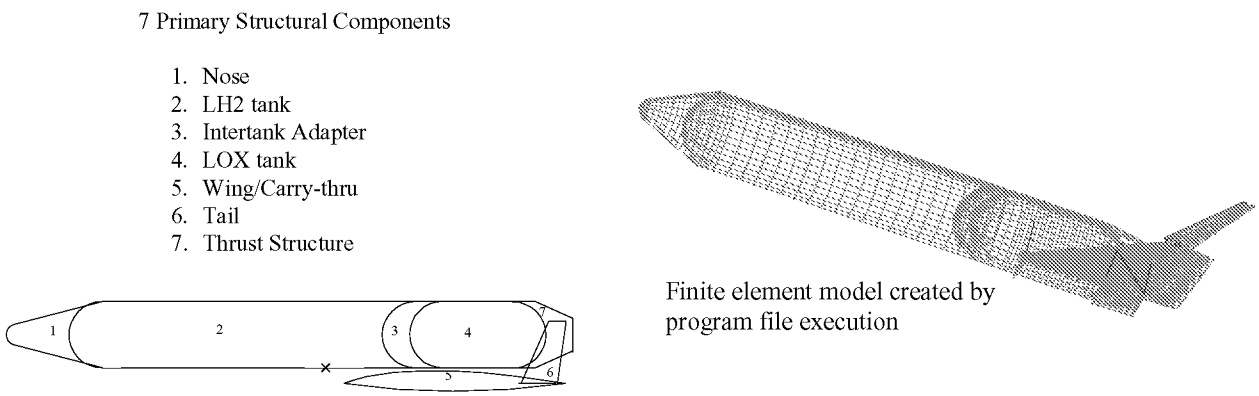


Figure 3) Structural Components for the example case and the associated analysis model

IDEAS program files have been created which will automatically generate the seven components of this vehicle based upon input geometric parameters. Program files are scripts of IDEAS GUI commands which include some programming level constructs like variable definition, looping and logic decisions. They enable IDEAS to be operated in a batch fashion using an input file of controlling commands. The FEA model shown in Figure 3 is the resulting model that is created by running the IDEAS program files which describe its seven structural components. These parts include NURB based geometric information as well as the required finite element mesh data, all of this entity creation is controlled in a part's template program file. Instances of those parts with appropriate input geometric information create a vehicle for analysis. It is recognized that common geometry is required to work in a multidisciplinary environment [7], however at this low level of analysis complexity it was felt that common geometry only need imply that vehicle geometric design parameters which define the proposed configuration also are utilized to create instances of the parametric IDEAS parts. By extracting appropriate part dimension data from a vehicle sizing program output file the information can be used for instancing of the parametric part program files. This implies that in a more general multidisciplinary environment CAD geometry for individual disciplines, if required, must be generated in a form required by that discipline. If a higher level language is used to obtain geometry parameters from the vehicle sizing program these variables can be appropriately modified to fit input requirements of the structural parts. Structural models do not have to be derived from OML aero surfaces which often in the case of launch/entry vehicles represents an outer surface of TPS with the structure being well inside this mold line definition. A good example of this is the previously described X-33, the structural cross-section of this vehicle was always offset and somewhat dissimilar to the encompassing aerodynamic surface.

Fuselage parts are assembled simply by applying appropriate part translations and rotations such that each part is defined in a single consistent global coordinate space. Appropriate part placement as well as automated meshing of the parts is included in the part definition IDEAS program files. An assembly program file is then executed which will remove duplicate nodes at part boundaries. The wing and tail components of figure 3 are assembled to the fuselage in another simple manner. Structural load paths between the physically separate wing and fuselage parts are created by rigid link elements. Drag link elements define axial compatibility, and vertical elements shear wing lift load into the side of the fuselage. Rigid link elements are similarly used to assemble the wing tip control fin to the wing at six locations between tip-fin spar and wing spar nodes. As only a 1/2 model is being assessed symmetry plane boundary condition requirements are automatically appropriately constrained. The stages of the entire automated process are outlined in Table II.

Table II – Steps Executed in the Automated Analysis Process

1) Parse CONSIZ output to determine input dimensions for the structural components.

2) Generate IDEAS program files from generic template program files and application of the parsed CONSIZ design data.

3) Execute the IDEAS program files to create separate structural components which define the vehicle parts.

4) Create the IDEAS assembly finite element model by eliminating duplicate nodes at part boundaries and assemble non geometrically similar components with rigid link elements.

5) Perform mass mapping of CONSIZ system weights to the finite element assembly model.

6) Create Basic Loadsets
   - ullage pressures
   - body and wing lift pressures
   - fuselage drag pressures
   - thrust forces (axial, wing-normal)
   - time dependent fluid head pressure loads
   - trim lift pressures
   - engine thrust and gimbal components (pitch direction only)

7) Parse the trajectory data file to obtain vehicle accelerations and fuel load parameters at design flight conditions.

8) For each design limit load condition create applied loadsets from the scaling and

superposition of the basic loadsets. Balance flight conditions in pitch by application of gimbal forces and aerodynamic trim control as required.

9) Restraint Set Creation, for each design loading condition create a corresponding restraint set

10) Create Internal element loads for each design loading condition, performed by execution of the FEA static solution for each design loading condition.

11) Structural Concept and Material Selection, Structural Sizing
- Run HyperSizer to determine Theoretical Structural Weights

12) Structural Component Processing
- For each structural component (tank, wing…) Process theoretical weight into As Built Component Weight

13) Vehicle Processing
- Process As Built component weights into CONSIZ vehicle weight input parameters

CONSIZ [14] is a VAB launch vehicle sizing program which provides a vehicle weight statement and geometry parameters. The goal of the entire automated structural weight estimation process is to refine the vehicle weight estimates made in CONSIZ and make them sensitive to structural and technology decisions as has been previously described. From a CONSIZ vehicle definition the IDEAS component program files can be generated and executed to create the FEA model shown in figure 3.

Typical FEA modeling requirements to define element property regions, loads, restraints and solver information are performed in items 5 through 10 of Table II. Step 5, mapping of masses from CONSIZ to IDEAS requires the use of a utility JAVA program which accesses data from both CONSIZ output and the FEA model. CONSIZ system masses are mapped to named groups of nodes defined in the IDEAS model or evenly distributed over a range of nodes between input fuselage stations. Step 6 creates basic loadsets which are combined with load scaling factors to create the combined loads fully representative of the design load cases listed in Table I. All load case solutions are simple linear static analysis conditions. To combine loads for a flight condition requires a balance of forces such that the vehicle is in a quasi-static state of applied external load. Non-flight conditions are restrained at nodal positions representing physical constraints and a fully balanced set of applied external loads is not required for a valid structural solution. For the flight conditions each of the basic loadsets are first analyzed in IDEAS to obtain applied load force and moment summations. These individual force summations are transferred to an Excel spreadsheet where the non-linear solver can be used to determine the amount of scaling required for balancing. As an example, for the Maximum flight acceleration condition, vehicle acceleration is known as is the amount of fluid in each cryogenic tank. To balance the axial acceleration the basic input thrust load is scaled, to balance vehicle rotation in the pitch plane the vehicle transverse thrust is also scaled. The axial and transverse thrust values calculated define the amount of engine gimbaling required for the quasi-static condition. For an entry condition the wing lift and control surface lift would have to be scaled to balance known vehicle normal acceleration while maintaining a zero pitch moment. Vehicle accelerations are assumed known and actually come from a POST trajectory analysis. Step 9 creates nodal restraints appropriate for each of the Table I design conditions. For flight conditions a node is still restrained but force summations for applied loads will be numerically zero at this point. Step 10 merely sets up required solution parameters and output file formatting of IDEAS universal files which will be in accordance with HyperSizer universal file input guidelines.

A Visual BASIC program is used to perform element sizing with HyperSizer. This program has been implemented on a parametric wing component and is now being incorporated into the full vehicle analysis process. The program has access to all of the HyperSizer COM (Component Object Model) objects defined in the HyperSizer Application Programming Interface (HyperSizer - API). The creation of an object model interface to HyperSizer via Microsoft COM programming techniques was also developed under NASA Langley's HPCCP program specifically to help automate the procedure being implemented here. Via this interface and the controlling BASIC program the IDEAS data will be read into a HyperSizer project, the sizing load cases of Table I will be setup for HyperSizer specific input data. For each structural region, defined by a unique property id number, structural configuration and panel sizing criteria are selected. These areas of unique property identification in IDEAS are termed components in HyperSizer and represent regions of constant wall geometric parameters and manufacturing technique. That is to say one HyperSizer component may be 2024 Aluminum metallic honeycomb sandwich construction, it will describe a small portion of the IDEAS structural component it is a part of, such as the IDEAS Nose component. There will be many HyperSizer components of this type within an IDEAS structural component and each will be permitted to vary geometry parameters as necessary to support applied loads for the input eleven design load cases. This gives the ability to tailor structural requirements to applied loading. For example consider the case of a LOX tank, where under axial acceleration head loads are considerably higher at the aft end then forward. The component breakup based sizing will necessarily tailor element thickness as a function of axial position in the tank so that a minimum weight IDEAS structural component weight will be calculated.

Both SDRC IDEAS, and HyperSizer have well defined data structures to quantify temperature dependent strength and stiffness characteristics of isotropic, and orthotropic materials. As HyperSizer is being used to perform actual element sizing the material property database within the program is used to maintain data required for analysis. HyperSizer outputs stiffness matrices to represent its stiffened element physical properties and these matrices are used by IDEAS for subsequent static analyses with a more representative set of element stiffness properties.

Upon completion of the automation procedure, results of HyperSizer element Sizing can be tabulated on a structural component basis and used to update mass estimation inputs to the CONSIZ program. Figure 4 shows one of the types of data presentation features of HyperSizer where Controlling loadcase is being reviewed for a sample input run. Graphical feedback to the user provides a means to check reasonability of a solution and is useful for presentation of results of a completed study. Element unit weight, failure mode and other structural performance parameters can be reviewed in the GUI environment once the automated process has been completed.

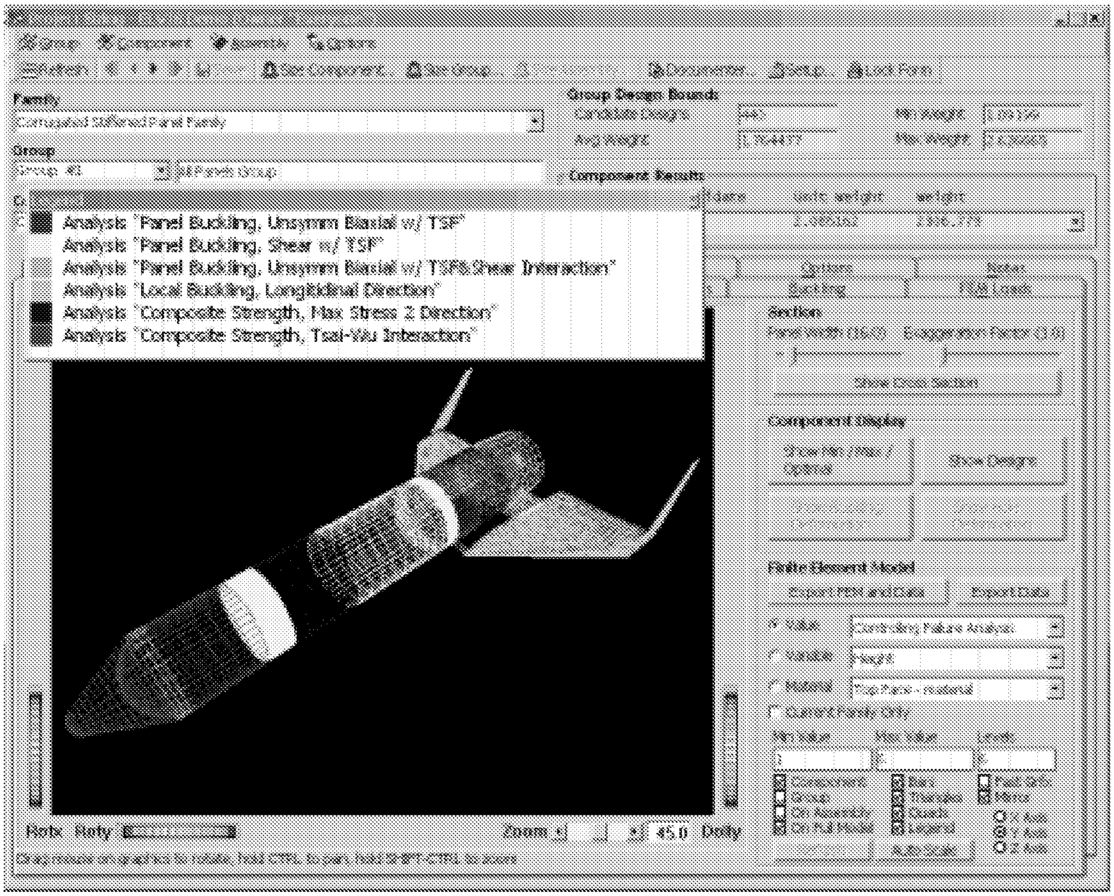


Figure 4) GUI presentation of HyperSizer Output, Controlling Loadcase

Ideally the process described above should be placed in an iterative loop, firstly to converge element load redistribution upon receiving element stiffness matrix updates from HyperSizer element sizing, and secondly to converge the iteration of providing new structural weight estimates to CONSIZ and it's effect on vehicle size. Figure 5 shows a flowchart representation of what has been presented with the appropriate looping processes indicated.

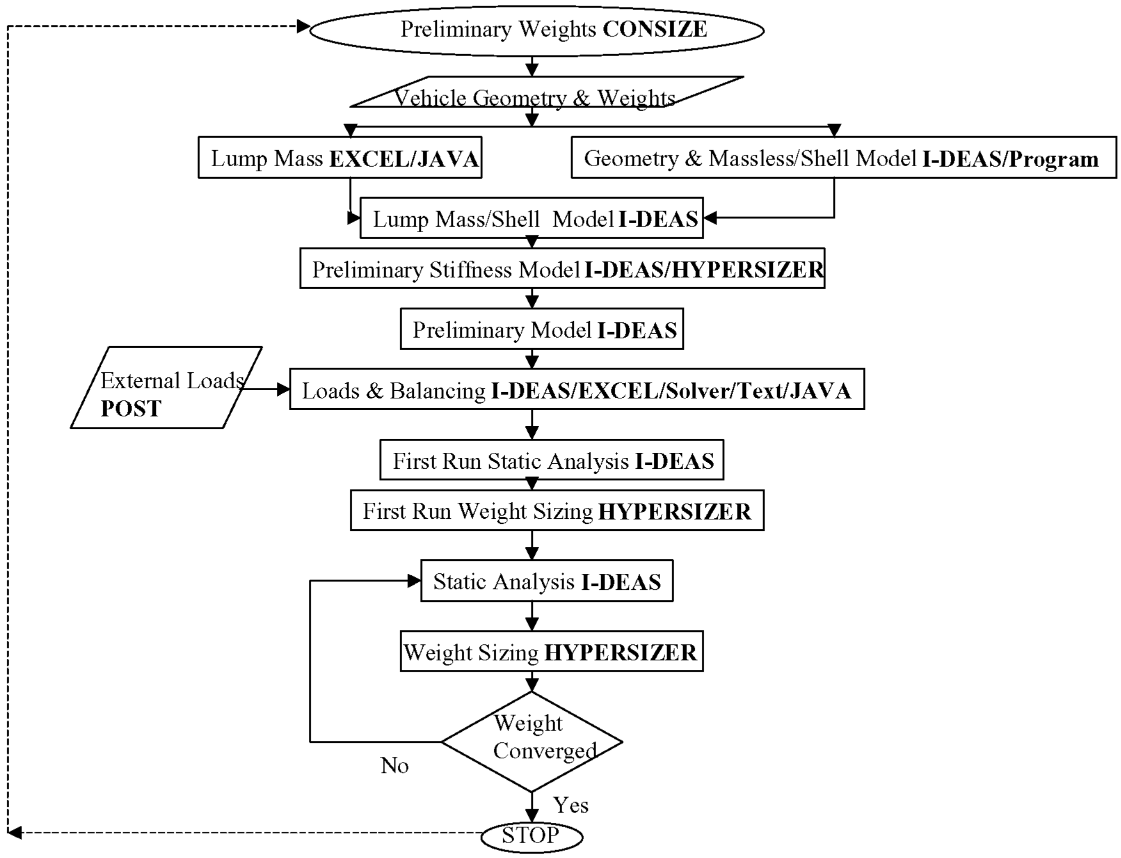


Figure 5) Flowchart representation of the automated structural weight estimation process

CONSIZ is typically run with the requirement of having a specific mass ratio. That is the propellant weight ratio of pounds of fuel at liftoff to pounds of fuel at Main Engine Cutoff, MECO. Having a vehicle sized just large enough to contain enough propellant to achieve the required mass ratio assures an achievable orbit. If the structural analysis process is allowed to update CONSIZ mass estimations the resulting vehicle size will be different than was initially calculated because of dry weight changes. Along with this overall vehicle design loop are aerodynamic and aerothermodynamic effects which are currently not captured in the structural sizing process. However by providing a set of automated procedures which can perform structural weight estimation a more encompassing vehicle design process may call upon the structural process as only one element of its overall optimization goals. One must keep in mind that the process just described permits variation in vehicle geometric design parameters and so opens up the overall vehicle design to allow variability in fuselage fineness, wing thickness and planform parameters and positioning of major structural components. Figure 6 shows the structural assessments and weight estimation methods interacting in a proposed full vehicle synthesis environment. This is a large suite of tools requiring data exchange. Often the codes run on different operating systems and are programmed in different computer languages. One method of integrating such an array of computer programs is to use a commercial framework product such as Model Center. By creating a Model Center process for the structural weight estimation procedures that have been described it is hoped the integration of those procedures into a full vehicle synthesis program can be obtained.

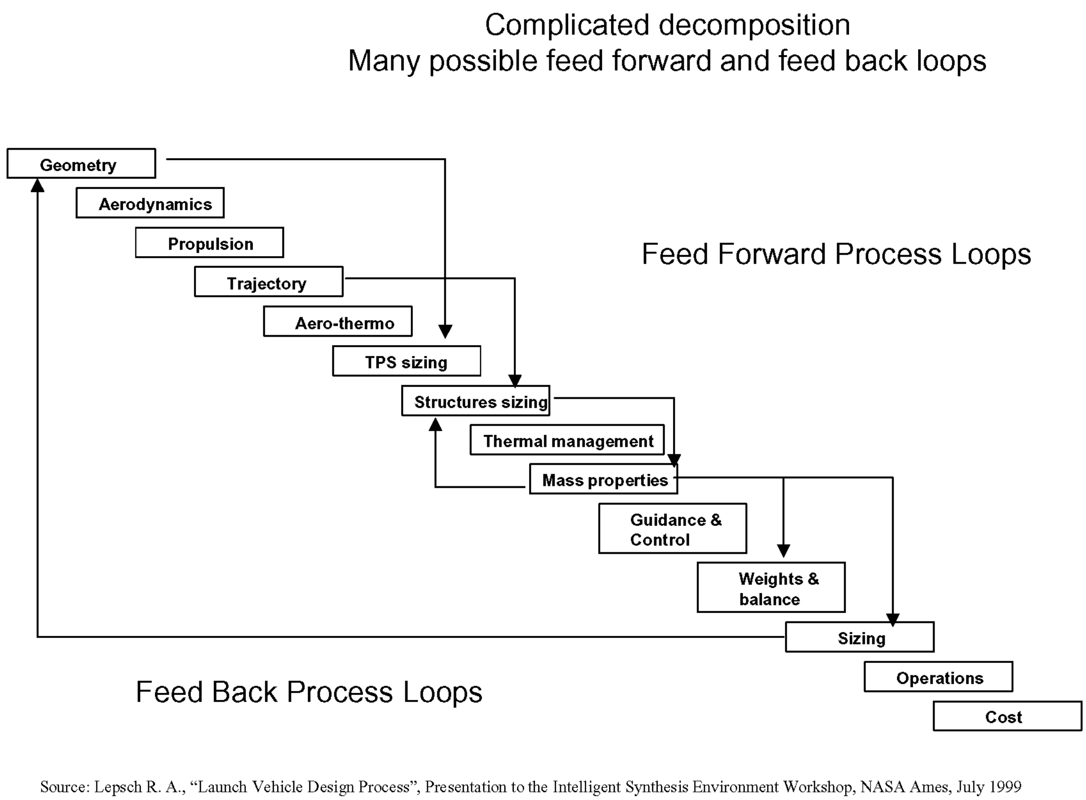


Figure 6) Process flow possibilities in a full vehicle synthesis system analysis tool

Framework Environment:

Model center allows each computer program to run as a server application on a decentralized computer. The program is then termed an "Analysis Server" component and it is available to a Model Center Client application which defines code interaction and process flow. The details of Model Center operations regarding how data is "wrapped" to feed from one analysis server component to another will not be fully explained here as they are defined in the Model Center Users Manual. Suffice it to say that legacy code, ASCII files, Excel spreadsheets and script language programming can all be used to create and manipulate data in the Model Center Framework.

Figure 7 shows the client application view of the Model Center wrapped structural analysis process. The main area of hierarchical text on the left portion of this screenshot is a view of the main process names, some breakout of main processes and some input/output data for a typical process. The entire process is titled "AdvancedStructuralAnalysisV2". Under that heading are three main subcategories, CONSIZ, IDEAS, and HyperSizer. These three subcategories are also shown in the topmost process flow diagram to the right of the text window. The second process flow window shows the sub processes that occur in the IDEAS process. This hierarchical representation of process flow can repeat as required until a projects entire process flow is defined. For example the process IDEAS/LoadsBalancing also contains sub processes. Although not depicted in the process flow windows (by user choice) they are shown in the hierarchical text window. There we see three sub processes, preLoadsBalancing, loadsBalancing, and postLoadsBalancing. The analysis steps outlined in Table II are implemented in the Model Center environment shown in figure 7. Some processes manipulate data to prepare it for use in a subsequent process, and some processes may only execute legacy code. As an example of legacy code execution many of the processes in the analysis merely call for a batch execution of the IDEAS program with an input filename. To actually create the data in the file that is being run in the IDEAS process is more complicated. Here JAVA code is often used to operate logically on the data from upstream processes and create the required data for the downstream process. Another complicated process is the LoadsBalancing process. This process is an example of using ASCII files and Excel spreadsheets to create input loads based on scaling the basic IDEAS loadsets to suit the trajectory acceleration and flight condition requirements of load cases 4 through 8 in Table I. A legacy spreadsheet process is integrated into the LoadsBalancing process and permits use of the spreadsheet solver function to help determine scaling factors for simultaneously balancing pitch trim, axial and wing-normal forces.

Use of a framework tool such as Model Center is very enabling in rigorously defining data and data processing requirements for a complex analysis system. It is easily envisioned that each analysis process shown in figure 6 could be implemented in a Model Center environment. This can be done by having the discipline expert create Analysis Server components on there own computer hardware. Discipline components are then available to fit into the larger process. From that baseline of analysis programs the entire vehicle synthesis wholly outlined in figure 6 can be implemented as another Model Center client application.

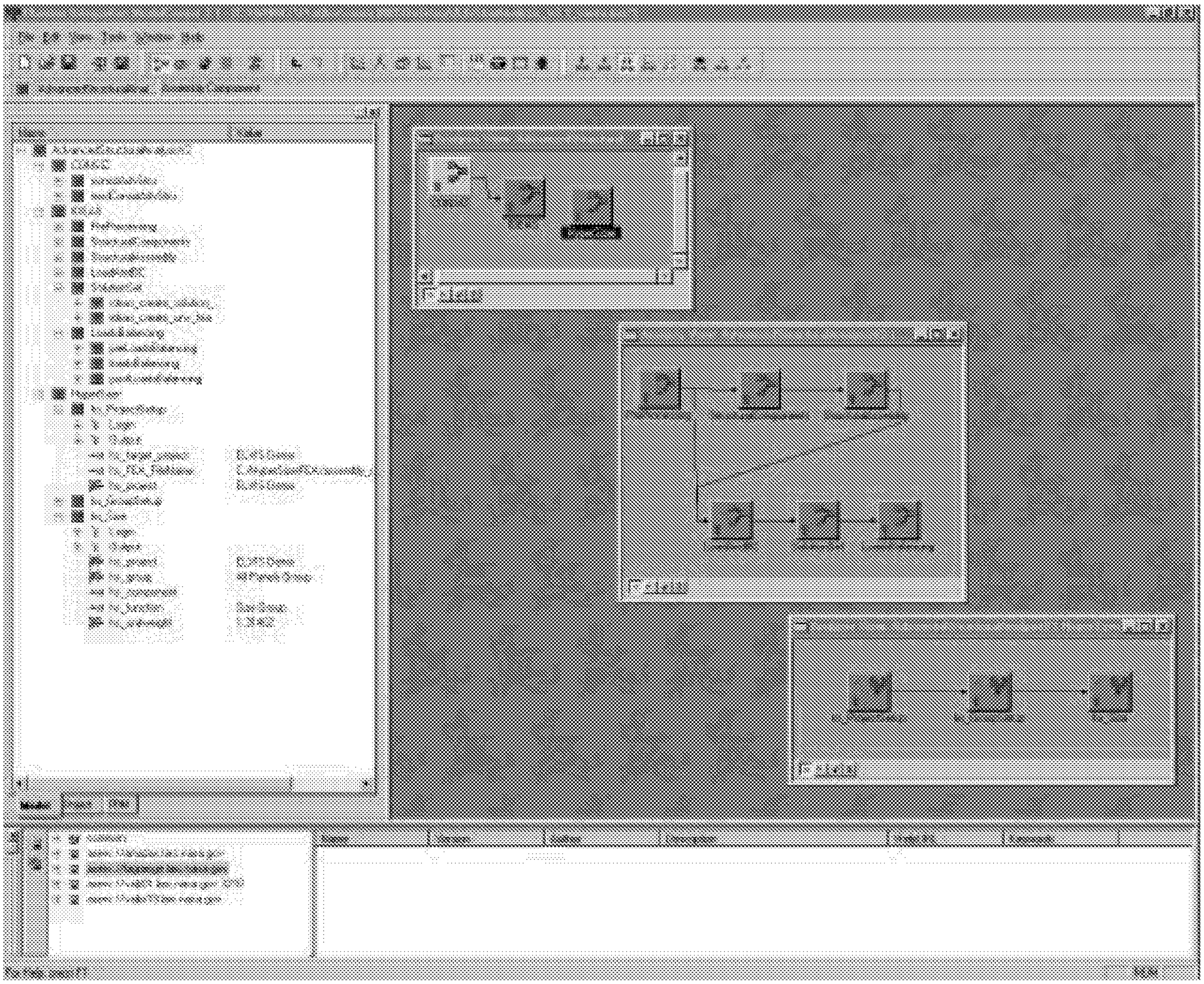


Figure 7) Model Center Process Flow for the FEA based Structural Weight Estimation Process

Conclusions and Future Plans:

A process under development at NASA Langley Research Center's Vehicle Analysis Branch which formalizes bottoms up structural weight estimation for launch vehicles has been described. The features of basing such a process on 3D finite element data structures are presented. The analysis task requires certain generic analysis steps to be implemented. Automatic execution of these steps is demonstrated and is accomplished using the IDEAS commercial program for CAD and FEA solution work. The commercial program HyperSizer is used to size finite elements and make theoretical component weight estimates. Utility code is used to manipulate process data between upstream and downstream stages of the analysis. These various stages of analysis may occur under differing operating systems and all processes are integrated via the commercial framework, Model Center. Automatic execution is desired so that structural weight estimation work can be incorporated into a multidisciplinary design loop and provide weight sensitivity to vehicle geometric parameters, structural arrangement, structural concept selection, material property selection, loading and failure mode criteria.

This development has been supported by NASA's High Performance Communications and Computing program (HPCCP) and is currently being supported by NASA's goal of developing an inter-center Advanced Engineering Environment (AEE) for launch vehicle architecture studies. Implementation of the required analysis stages is demonstrated for a Single Stage to Orbit (SSTO) vehicle. It is shown that because of the general capabilities of the tools utilized and the utility code developed to manipulate data for these tools more general vehicle configurations and structural designs can be accommodated.

One item that has not been discussed in this paper is calibration of the weight estimation procedure to existing vehicle weights. Some of this work has actually been done on a single structural component such as a vertical tail. However to build confidence in the system it must be exercised against existing systems such as Atlas, Delta, and the Space Shuttle. This should be a goal for future work. The process has been well defined, only specific CAD models for unique structural elements in this system need to be generated and assembled appropriately. Sufficient generality exists in the procedure to perform validation on all of these types of vehicles provided sufficient mass properties, configuration, and design load information can be obtained.

References

[1] Taylor, A.H., Jackson L.R., Cavis R.C., Cerro J.A., and Scotti S.J: “Structural Concepts for Future Space Transportation System Orbiters”, Journal of Spacecraft and Rockets, May-June 1985, V22, No. 3.

[2] Taylor, Allan H.; Cerro, Jeffrey A.; Cruz, Christopher I.; Jackson, L. R.; Naftel, J. C.; and Wurster, Kathryn E.; “Orbit on demand - Structural analysis finds vertical launchers weigh less”, Aerospace America, February 1985.

[3] Dorsey J.T., Myers D.E., and Martin C.J.: “Reusable Launch Vehicle Tank/Intertank Sizing Trade Study”, AIAA Paper No. 2000-1043.

[4] Bush L.B., Lentz C.A., Robinson J.A., and MacConochie I.O.: “Structural Design Considerations for a Personnel Launch System”, Journal of Spacecraft and Rockets, Vol. 29, No. 1 1992.

[5] Anderson D. M., “NASA’s Integrated Space Transportation Plan”, AIAA Paper 2000-3828

[6] Holcomb L.; Smith, P. and Hunter, P.: “NASA High Performance Computing and Communications Program”, NASA-TM-4653, 1994.

[7] Yanez, D. P.; “Rapid finite element modeling with RAMPAGE (Rapid Aircraft Mass Properties And Geometry Estimation)”, SAWE paper No. 2465, 1999.

[8] Space Propulsion Synergy Team, “A Guide for the Design of Highly Reusable Space Transportation”, NASA/Industry Consortium, August 1997.

[9] Ardema, Mark D., “Body Weight of Hypersonic Aircraft: Part 2”, NASA TM 102797, August 1990.

[10] Welch S.R.: “PDWING – A Preliminary Design Multi-Station Wing Analysis Program”, SAWE paper No. 1971, 1990.

[11] IDEAS, commercial CAD/CAE software, available from, EDS Co. 5400 Legacy Drive, Plano, Texas 75024-3199.

[12] HyperSizer, commercial CAE software, available from, Collier Research Co. 45 Diamond Hill Rd. Hampton, VA 23666

[13] Model Center, commercial computer software program, Phoenix Integration Co. 1715 Pratt Drive, Suite 2000, Blacksburg, VA 24060.

[14] Lepsch, R. A., Jr., Stanley, D. O., Cruz, C. I., and Morris, S. J., Jr., “Utilizing Air-Turborocket and Rocket Propulsion for a Single-Stage-to-Orbit Vehicle,” Journal of Spacecraft and Rockets, Vol. 28, No. 5, 1991, pp. 560-566.