IMECE2011-64510

AN INTERACTIVE DESIGN SPACE SUPPORTING DEVELOPMENT OF VEHICLE ARCHITECTURE CONCEPT MODELS

Gary Osborne
University of Louisville
Louisville, KY, USA

Glen Prater
University of Louisville
Louisville, KY, USA

Rostyslav Lesiv
University of Louisville
Louisville, KY, USA

David Lamb
U.S. Army RDECOM-TARDEC
Warren, MI, USA

Matthew Castanier
U.S. Army RDECOM-TARDEC
Warren, MI, USA

ABSTRACT

Due to a lack of suitable analysis tools, automotive engineers are often forced to forego quantitative optimization early in the development process, when fundamental decisions establishing vehicle architecture are made. This lack of tools arises because traditional analysis models require detailed geometric descriptions of components and assembly joints in order to yield accurate results, but this information is simply not available early in the development cycle. Optimization taking place later in the cycle usually occurs at the detail design level, and tends to result in expedient solutions to performance problems that might have been more effectively addressed at the architecture level. Alternatively, late-cycle architecture changes may be imposed, but such modifications are equivalent to a huge optimization cycle covering almost the entire design process, and require discarding the detail design work used originally as the basis of the NVH model. Optimizing at the architecture level can both shorten and improve the results of a vehicle development process. In this paper we describe the requirements and implementation of a user interface for a software package supporting vehicle architecture conceptual design and analysis.

INTRODUCTION

By dividing a vehicle structure into connected functional assemblies and assemblies into functional components - beams, surfaces, major compliance joints, and assemblage joints - and modeling those components in a simple, direct fashion, it is possible to develop an attribute-based first-order model for a vehicle. These attribute-based models are smaller than traditional models, straightforward to modify, and because of the division into functional components, simple to interpret. We shall refer to simplified attribute-based models as "concept models[1]," By including abstractions specific to engines, motors, transmissions, differentials, power split devices, transfer cases, fuel tanks, batteries, and brakes, concept models can accurately predict the inertial properties, compartment volumes, clearances, top speed, maximum acceleration, minimum braking distance, fuel efficiency, payload capacity, structural integrity, and NVH characteristics of a vehicle architecture without requiring a comprehensive geometric description. They can be used to optimize the architecture layout of a vehicle, conduct iterative design studies, or develop reference models based upon a baseline design.

This paper describes the requirements and application of a user interface for a software package supporting vehicle architecture design and analysis. The software, CMTS (Concept Modeling Tool Suite), permits specification of function-based abstractions that decouple architecture design from the geometric CAD models that form the basis of most traditional analysis models. This decoupling enables the software to function as a "design space for ideals." Along with an interface for specifying and modifying an architecture-level vehicle model, CMTS includes a range of complementary analysis modules (calculation of geometric and inertial parameters, rigid body response, structural FEA, including NVH characteristics, powertrain performance, stability, ergonomic assessment, etc.)

---

[1] The nomenclature "simplified model" has also been applied to attribute-based FEMs. We avoid this terminology because these models, while small in terms of element count, involve modeling decisions critical to the overall accuracy of the results. In fact, the use of specialized elements and joint representations add a level of complexity not present in detailed models.

Report Documentation Page

1. REPORT DATE **17 JUN 2011**	2. REPORT TYPE **Journal Article**	3. DATES COVERED **17-06-2011 to 17-06-2011**

4. TITLE AND SUBTITLE **AN INTERACTIVE DESIGN SPACE SUPPORTING DEVELOPMENT OF VEHICLE ARCHITECTURE CONCEPT MODELS**	5a. CONTRACT NUMBER **w56hzv-04-c-0314**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S) **David Lamb; Matthew Castanier; Rostyslav Lesiv; Glen Prater; Gary Osborne**	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Louisville,Louisville,KY,40802**	8. PERFORMING ORGANIZATION REPORT NUMBER **; #21931**

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) **U.S. Army TARDEC, 6501 E.11 Mile Rd, Warren, MI, 48397-5000**	10. SPONSOR/MONITOR'S ACRONYM(S) **TARDEC**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) **#21931**

12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES **Submitted to ASME 2011 INTERNATIONAL MECHANICAL ENGINEERING CONGRESS AND EXHIBITION(IMECE)**

14. ABSTRACT **n/a**

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**10**	

appropriate for characterizing the architecture-level performance of a vehicle concept.

REVIEW OF PREVIOUS LITERATURE

In the automotive industry, the original first order concept models were largely discarded in favor of finely meshed shell element full-body models after the advent of high powered workstations. However, the advantages of concept models are so compelling that designers, researchers, and analysts are revisiting their use. The advantages of concept models (referred to as "hybrid models"), based upon beams and shell elements, has been described and correlation with experimentally measured parameters was undertaken, with good results [1]. By using concept models and detailed models in support of a passenger car development program, NVH improvements and reduced development time was possible [2]. Shortcomings in detailed FEM such as long modeling time and lack of detailed architectural features required for an accurate model result in critical design decisions being made without CAE support [3, 4]. These investigators used parametric topology/concept models to conduct stochastic studies that yield an optimized conceptual design, which then serves as a starting point for intermediate and detail design.

Concept modeling methodologies have been integrated with a goal programming optimization algorithm [5]. The very critical issue of representing major body joint compliance in architecture concept models was addressed by a number of works [6-8]. Suitability of using concept models for pickup truck boxes was investigated [9]. Beam-only concept models were used to support the design of a construction vehicle cab [10]. Beam/shell FE concept models have been used to reduce weight and increase stiffness of a light-duty truck floorpan [11].

While some researchers have developed parametric based concept models with automated FE meshing [12, 13], it has not previously been done at the vehicle level, nor has the additional analysis tools and stages been encompassed into the concept development process. Our work here addresses a number of open issues, including abstractions of concept modeling techniques for various types of vehicle performance and architecture assessment completed during conceptual design.

VEHICLE CONCEPT MODELING OVERVIEW

Vehicle architecture abstractions suitable for comprehensive conceptual design assessment involve the representation of functional assemblies and their association to one another including structural connectivity and energy/power transmission paths. Each assembly may be categorized as structural, inertial, energy storage, power source, or power transmitting. All assembly types require a position transformation including a vector description of the assembly's coordinate system relative to the vehicle coordinate system, and orientation within the vehicle coordinate system based on a set of three Euler angles. Each assembly type requires structural connectivity information to at least one other assembly within the vehicle architecture and all assemblies within the vehicle

must be interconnected in some manner to form a contiguous vehicle. Any energy storage, power source, or power transmitting assembly needs additional paths to represent energy and power flow within the vehicle. Figure 1 depicts a light-duty rear wheel drive four door pickup truck architecture with structural ladder frame, crew cab compartment, and payload assemblies, fuel tank energy storage assembly, V8 SI engine power source assembly, and transmission, differential, and wheel power transmission assemblies.
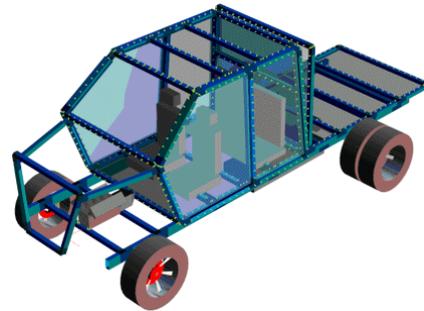

Figure 1. Pickup truck architecture abstractions.

Structural Assemblies

Assemblies containing components that may deform significantly relative to the amount of deformation in the assembly connections when external loads are applied to the vehicle are termed structural assemblies. Concept modeling methodologies for structural assemblies begin with a functional division of architectural features based on the internal load distribution, followed by an assessment of how critical the influence of a component's features are on the overall model performance. A structural assembly concept model must represent the architecture layout and connectivity involving primary load carrying structural members (beams), major compliance joints (MCJ) at the beam junctions, panels carrying secondary in-plane loads, inertial items, and assemblage joints connecting components to other components, Figure 2. These component and connection type abstractions are sufficient for constructing structural assembly concept models.

Beams Many of the primary structural members in most vehicle body assemblies are beam-like structures capable of carrying various combinations of axial forces, bending moments, and torsional moments. These beam components are characterized by a length much greater than the width and depth of the cross section. Beams are characterized by the path formed by the locus of cross sectional centroids along the length of the member (drag path), cross section shape defined by the median line of the sheet metal in the cross section, and sheet metal gauge at various locations within the cross section, Figure 3. The drag path of a beam may be simple for a straight member with constant section, or quite complex for a curved member with multiple section changes.
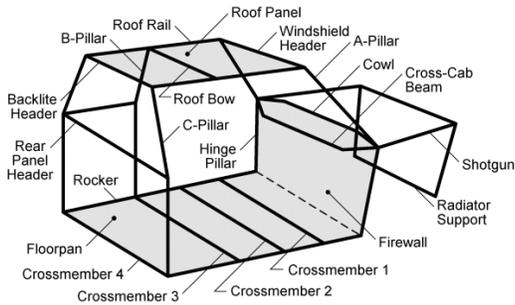
Figure 2. Pickup truck crew cab compartment structural components abstractions.
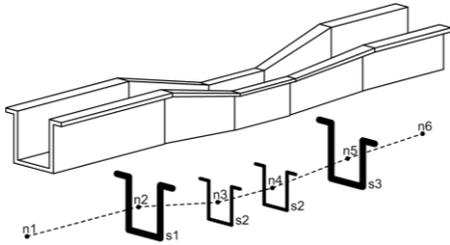

Figure 3. Beam abstraction including 6 GCPs, a piece-wise linear drag path, and 5 cross section regions (3 constant and 2 tapered.)

<u>Major Compliance Joints</u> Junctions of two or more beam-like members in a vehicle body structure can be modeled as MCJs. These joint types are often quite flexible in at least one direction, and their compliance permits relative rotation among the intersecting beam branches. The magnitude of this compliance is large enough that such joints have a significant effect upon all aspects of a vehicle body's static and dynamic response. Furthermore, MCJ characteristics are strongly influenced by local topology, sheet metal gauge, and assemblage joint details, and thus are a target of design optimization efforts.

The best method for modeling MCJs involves sets of elastic parameters for the individual beam branches [14]. The elastic parameters for each leg includes one parameter for angular deflection about the legs centroidal path and two additional parameters related to orthogonal angular deflections along the leg's path. These parameters are used as scaling factors for the torsional constant and area moments of inertia.

<u>Panels</u> Most auto body architectures contain secondary shell-like members with large flat or slightly curved surface areas and very thin wall thicknesses capable of carrying in-plane loads through strain energy storage. Panel geometry is characterized by a surface boundary, prominent interior features, and wall thickness, Figure 4. The surface boundary may be simple for a flat rectangular panel or quite complex for a curved surface with cut-outs, stamp-in beads, and a highly curved boundary.
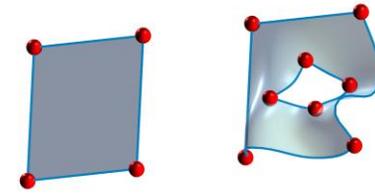

Figure 4. Panel component abstractions.

<u>Inertial Components</u> Nearly all assemblies contain components that are not designed to carry structural loads in the assembly, such as seats and other trim items. However, these inertial items have an influence on the dynamic response of the assembly and overall vehicle. The critical parameters describing the inertial components contributions to the assembly are the inertial parameters and their attachment locations. Inertial parameters include the mass of the item, centroid, and mass moments of inertia. These inertial parameters can be specified directly or determined from the geometric abstraction representing the component. Options for the geometric representation include path with cross sectional area, surface with thickness, and enclosed volume with specified inertial parameters. When specifying a volume, all inertial properties for the component must be specified since the internal material distribution is still unknown in nearly all cases except the trivial solid homogeneous part. Enclosures are represented by parametric equations as well and they are defined by a closed set of bounding surfaces.

<u>Assemblage Joints</u> The majority of the spot welds, adhesive bonds, or fasteners in a vehicle body structure can be modeled as assemblage type joints. These assemblage joints occur between beams, panels, and inertial components and are accurately modeled by a set of rigid connections at the corresponding physical fastener parametric locations relative to the component. There are three assemblage joint classifications; point, path, and surface based connections with each one defined by the geometric relationship between two components, Figure 5.
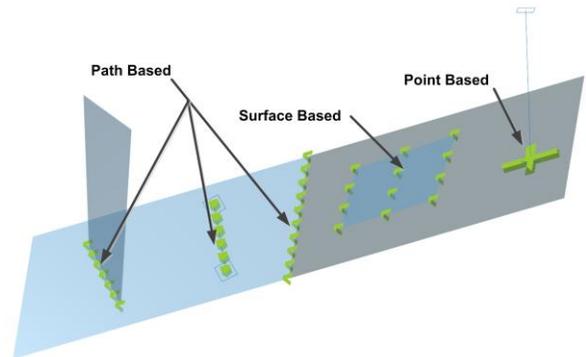

Figure 5. Assemblage joint classifications.

Point based connections occur between a beam component intersecting a panel component and rigid components connecting to other rigid components at a joint. Point based connections involving panels are modeled by a single multi-

point connection that fixes the beam centroid to a set of corresponding connection locations on the panel. These multi-point connections help evenly distribute the large out of plane loads that may be transferred to the panel by the beam. The only data abstraction required for point based panel component connections are the quantity of attachment points to the panel and the radius of the attachment points. Point based rigid-to-rigid component connection abstractions require specification of the joint's DOF if any exist.

Path based connections occur between beam components tangential to panel components and two panel components sharing an edge dependency. Data abstractions for path based component connections include weld pitch spacing and parametric range along the connection path. This level of information is sufficient to define component discretization points parametrically.

Surface based connections occur between two panel components partially sharing a surface and panel component to enclosure component connections. They simply extend the path based connection to a second dimension by adding an additional weld pitch spacing and parametric range in the orthogonal parametric coordinate. Similar to path based connections, component discretization is done parametrically based on the connection data.

Inertial Assembly Abstraction

Any assembly that does not specifically provide structural support for the vehicle architecture can be modeled as an inertial assembly. The critical features of these inertial assemblies are the inertial properties and the connectivity to the other assemblies in the vehicle. Mass and mass moments of inertia for the assembly are the only abstractions required to define the concept model. However, specialized abstractions are required for critical functional assemblies influencing powertrain performance such as acceleration, braking, and fuel efficiency. These powertrain assemblies can be classified into three functional groups; energy storage, power source, and power transmission assemblies.

Energy Storage Assemblies   Fuel tanks, batteries packs, and hydraulic accumulators in a vehicle are examples of energy storage assemblies, Figure 6. Each energy storage type provides a unique form of energy for a specific type a power source assembly. Abstractions appropriate for these energy storage assemblies require parameters to determine the amount of available energy and approximate estimates of their inertial properties based on typical shapes.

Power Source Assemblies   Internal combustion engines, electric motors, hydraulic accumulators, and brakes are examples of power source assemblies within a vehicle, Figure 7. Brakes are considered as power source assemblies if they are implemented as regenerative assemblies for either electric or hydraulic vehicles. If not, they are power dissipating assemblies. These assemblies convert stored energy into work to propel the vehicle or generate energy for some electric/hydraulic hybrid systems. Energy consumption or

production rates, torque outputs, and inertial properties must be captured by the abstractions of these assemblies.
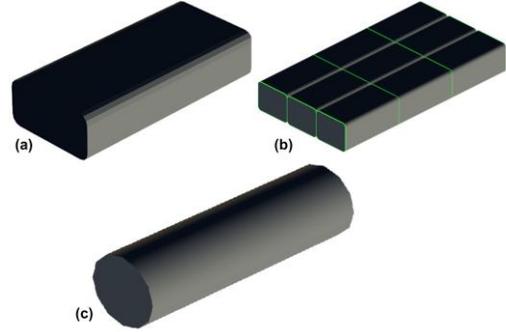

Figure 6. Energy storage assemblies a) fuel tank, b) battery pack, and c) hydraulic accumulator.
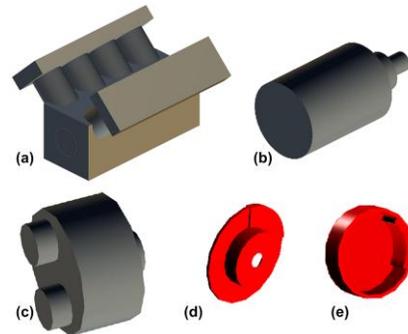

Figure 7. Power source assemblies a) V8 IC engine, b) electric motor/generator, c) hydraulic motor, d) disc brake, and e) drum brake.

Power Transmitting Assemblies   Power split devices, transmissions, differentials, and wheels are a few examples of power transmitting assemblies, Figure 8. These assemblies transmit power from one assembly to the next assembly or assemblies connected by a power transmission path or paths. For wheels; power at a driven wheel is transferred to the ground. In addition to abstractions for the inertial parameters, speed and torque input to output ratios and proportioning of output are the critical abstraction parameters for vehicle assessments.
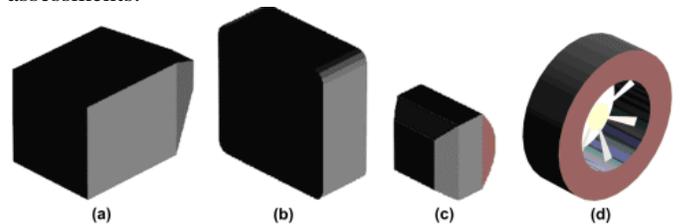

Figure 8. Power transmitting assemblies a) transmission, b) power split device, c) differential, and d) wheel.

Assembly connections

Structural connections among assemblies have a significant impact on the dynamic characteristics of a vehicle's architecture. The critical features of these structural assembly connections involve the geometric locations of the attachment

points within the vehicle along with individual connection properties. These connection properties include the DOF, compliance, and damping properties associated with each attachment in the assembly connection. An assembly connection is represented as a set of connections between two assemblies contained in a vehicle concept model.

Energy and Power Transmission Paths

These connections are normally insignificant in terms of the structural loads supported by the vehicle architecture itself, especially the energy transmission paths such as fuel, electric, and hydraulic fluid lines. The loads carried by shafts to transmit torque between the aforementioned power source and power transmitting assemblies have a significant impact on the shaft design but they can readily be designed in isolation of the vehicle architecture. The primary purpose of these path abstraction types for the vehicle architecture are the relationships of energy consumption by power source assemblies from connected energy storage assemblies and speed/torque changes that occur as power passes through power transmitting assemblies. Figure 9 depicts energy flow (yellow for fuel and green for electric currents) and power transmission (red) for a conventional and parallel electric hybrid powertrain.

CMTS APPROACH

Vehicles are developed based on a set of functional requirements such as number of occupants, quantity of payload volume/weight, acceleration and braking performance, NVH, and crash energy dissipation, etc. By dividing a vehicle into subassemblies with parameterization based on functional intent, it is possible to develop a full vehicle concept model that can be assessed in terms of the functional requirements at the vehicle level as well as the subassembly level. Furthermore, it is possible to capture fundamental design information related to how the vehicle is configured by defining a hierarchy of the subassemblies.
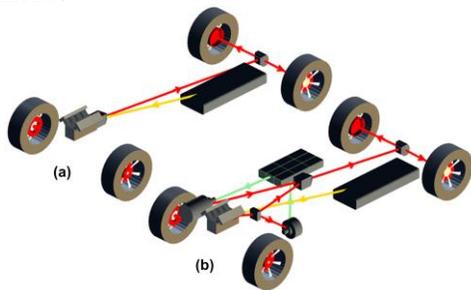

Figure 9. Vehicle energy and power transmission paths for a) conventional IC engine and b) parallel electric hybrid rear wheel drive powertrains.

The CMTS user interface for the concept vehicle development software follows a hierarchy that represents the various levels of the vehicle. The top level is the entire vehicle which defines the types of assemblies within in the vehicle, their position and orientation within the vehicle, and the connectivity between one another. Below the vehicle level is the assembly level, which defines the architecture layout of an individual assembly by defining component positions, orientations, and connectivity to one another. Below the assembly level is the component level, which is where component data is specified such as member sizes, gauges, and construction material.

Analyses may be performed at the vehicle, assembly, or component level, but the type of analyses which may be executed depends on the current level in the hierarchy. For instance, at the component level, only inertial and structural analysis is possible, but at the assembly level compartment volumes and ergonomics may be assessed for assemblies containing crew members. At the vehicle level, there are many more analyses such as rigid body dynamics and a host of powertrain performance analyses. Thus, it is possible to develop and assess individual assemblies for inclusion into a vehicle one assembly at a time.

The user interface breaks the three levels into three specialized window types. A main window that contains a vehicle hierarchy pane with options for adding, editing, and arranging various types of assemblies into the current vehicle provides a means to layout vehicle level architecture. Assembly architecture layout is controlled in the assembly window permitting the designer to add, edit, and arrange beam, panel, and inertial components. A component window accessed through the assembly window permits the designer to specify information specific to a particular component within an assembly. While the software interface is divided into this hierarchal representation of the vehicle, this paper will discuss the key factors supporting this vehicle hierarchal model.

VEHICLE HIERARCHY

This vehicle hierarchy describes the type of subassemblies comprising a vehicle, their connectivity, and the critical attributes of the subassembly related to vehicle performance. As an example, Figure 10 depicts a vehicle hierarchy for the light-duty pickup truck shown in Figure 1 on the left and a typical cab-over-engine (COE) tractor trailer on the right.

From the pickup truck hierarchy, it is clear the pickup truck contains a ladder frame, cab crew compartment, two suspensions, four brakes, four wheels, two differentials, transfercase, transmission, engine and fuel tank. The chassis includes a McPherson strut front suspension (nearest the top of the hierarchy based on axle position along the longitudinal axis of the vehicle) with disc brakes for stopping the wheels and a live axle rear suspension with ABS controlled drum brakes for stopping the dual rear wheels. Both front and rear suspensions incorporate a differential and a transfer case is defined under the ladder frame indicating the truck is four wheel drive. It has a 6-speed automatic transmission driven by a V8 spark ignition engine. Additional information embodied in the crew cab compartment abstraction could be included into the hierarchy such as number of seats and doors and architecture type such as engine forward with shotguns, engine forward, or cab over engine.
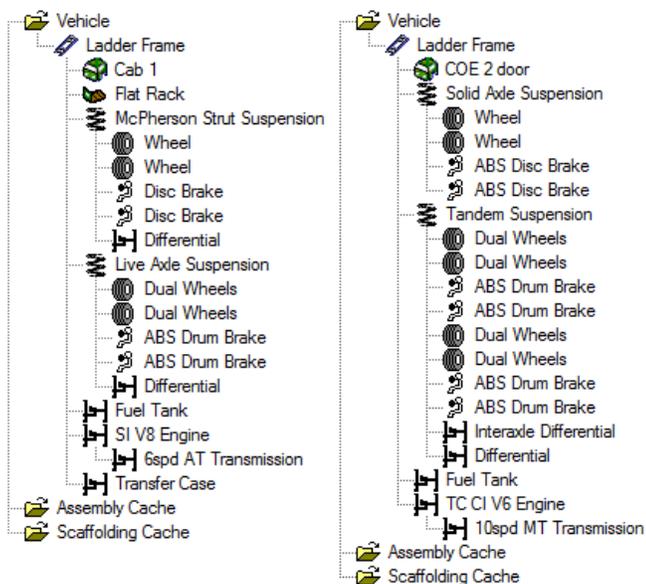
Figure 10. 4WD pickup truck architecture hierarchy on the left and COE tractor hierarchy on the right.

The tractor trailer vehicle hierarchy displays ABS wheels throughout with disc brakes on the front axle and drum brakes on the two rear axles connected through a tandem suspension. The tandem suspension contains both an interaxle differential and normal differential indicating that both rear axles are driven. The power plant is a turbo charged compression ignition V6 engine that connects to the driven axles through a 10-speed manual transmission.

Both hierarchies in Figure 10 indicate that the frame is the root (primary) assembly through which everything ultimately connects. The root assembly also establishes the vehicle coordinates system as the same coordinate system as itself. Items descending directly from the root assembly are connected to the root assembly by some assembly connection mounts. Similarly, any assembly below an assembly connects to the assembly from which it is a descendant. For the pickup truck hierarchy, the cab, flat rack, front and rear suspensions, fuel tank, engine, and transfer case all connect to the frame, but the wheels, brakes, and differentials connect to the suspension's axle housing which in turn is connected to the frame maintaining connectivity continuity. The transmission could be connected to the crew cab compartment as well, and could be represented in the hierarchy by creating an additional descendant line on the right side of the hierarchy. Most components only connect to a single parent assembly and thus the hierarchy model is sufficient for representing the connectivity network of the vehicle.

Assemblies are added to the vehicle by specifying the scaffolding, component, connectivity, and any additional abstract information of the specific assembly type and specifying which assembly it is a subordinate of. The first assembly defined for a generic vehicle type is always the root assembly. Appended to the vehicle hierarchy at the bottom are assembly and scaffolding caches. The assembly cache permits the designer to store alternative assembly architectures such as the crew cab compartment with the vehicle. These cached assemblies may be swapped with similar assemblies in the current vehicle design to optimize vehicle performance or investigate major platform changes. As an example for the pickup truck, a four door five passenger cab could easily be swapped with a two door three passenger cab to investigate vehicle performance based on the same chassis. The scaffolding cache holds any geometric models for any alternative assembly architecture that are not ready for the full component specifications.

Assembly Architecture Layout

The functional architecture layout is generated using scaffolding primitives to define paths to carry the primary loads, surfaces to enclose compartments and carry secondary in-plane loads, and enclosures to represent volumes or space occupied by abstract inertial items. Once these scaffolding primitives are specified, component specification including material, beam section shapes, panel thicknesses, and inertial properties for the nonstructural components commences to define the physical component model forming an assembly.

Scaffolding Points, paths, surfaces, and enclosures are the primitive geometric definitions required to specify the assembly architecture layout representing the functional intent of the structural components, Figure 11. Paths may represent nonstructural geometry in the model, boundaries of surfaces, or centroidal paths of beam components. Surfaces may depict nonstructural geometry, boundaries of volumes, or panel components. Enclosures characterize volumes occupying space and can contain specialized abstract nonstructural items with inertia such as those outlined in the vehicle concept modeling abstraction section. These geometric primitives are represented in the parametric domain to enable dependent relationships among geometric primitives within the model. These dependency relationships among the geometric primitives are critical to ensure model continuity when modifying a geometric primitive in the model. For example, if one were to modify the curvature of the truck cab roof about the plane of symmetry, one would expect that the roof bow curvature would update to follow the new roof curvature for adequate support of the roof.

Dependent geometric primitives are defined relative to other geometric primitives by the parametric dimensionality. The parametric dimensionality of the geometric primitives is zero, one, two, and three for points, paths, surfaces, and enclosures, respectively. Realizing that at least one parameter is required to define the position of one item relative to another, it is possible to specify items in terms of paths, surfaces, and enclosures only. Additionally, for the geometry of a dependent primitive to follow the shape of another item, its parametric dimensionality must not exceed the parametric dimensionality of the item which it depends upon. Thus, geometric primitives in the software include points dependent on paths, surfaces, or enclosures; paths dependent on paths, surfaces, or enclosures;

surfaces dependent on surfaces or enclosures; and enclosures dependent on enclosures. For the purpose of this article only the most utilized dependent primitives will be discussed.

As an example of the geometric primitives and their roles in defining the architectural layout of a structural assembly, the steps to create a four door pickup truck cab are discussed and illustrated in Figure 11.

a) Independent points establish the fundamental assembly shape and are specified by their Cartesian coordinates in the assembly's coordinate system.

b) Paths connecting the independent points define the outer perimeter boundaries of the assembly. By moving an independent end point of the path, the path will extend to the new location of the moved point. Additional internal control points for the path may be specified to define path curvature.

c) Dependent points specified on a path by a parametric location along the path enforce a dependency on the parent path. Thus, any geometric modification of the parent path (end point or curvature change) results in the dependent point moving in the Cartesian coordinates to maintain its parametric location on the parent path.

d) Paths connecting dependent points provide potential load paths for supporting other major architecture features. As with the paths before, their curvature may be specified with additional internal control points but their dependent end points will move in relationship to the parent paths or surface of each end point.

e) Surfaces connect a set of bounding paths joining end to end to form a closed loop. Their geometry depends upon the parent paths defining the boundary and any changes to the bounding parent paths result in a change to the surface geometry. Similar to paths, surface curvature away from the boundary is defined by additional internal control points.

f) Dependent paths connect dependent points on a path, surface, or the bounding parent paths of a surface. The curvature of these dependent paths is dependent on the geometry of the parent path or surface. Thus, their curvature changes with modifications of the curvature of the parent path or surface and not internal control points.

g) Dependent surfaces connect dependent paths of a single surface to form a surface dependent on the parent surface of the dependent bounding paths. Their curvature is entirely dependent on the parent surface curvature that they span.

h) Enclosures connect a set of bounding surfaces joining edge to edge to form a closed volume. Geometry of the enclosure is dependent on the bounding parent surface geometries and any modification to a bounding parent surface results in an alteration to the volume geometry.

i) Engine compartment enclosure, door bounding paths and surfaces, and model constraints (orange lines) to complete the scaffold model.
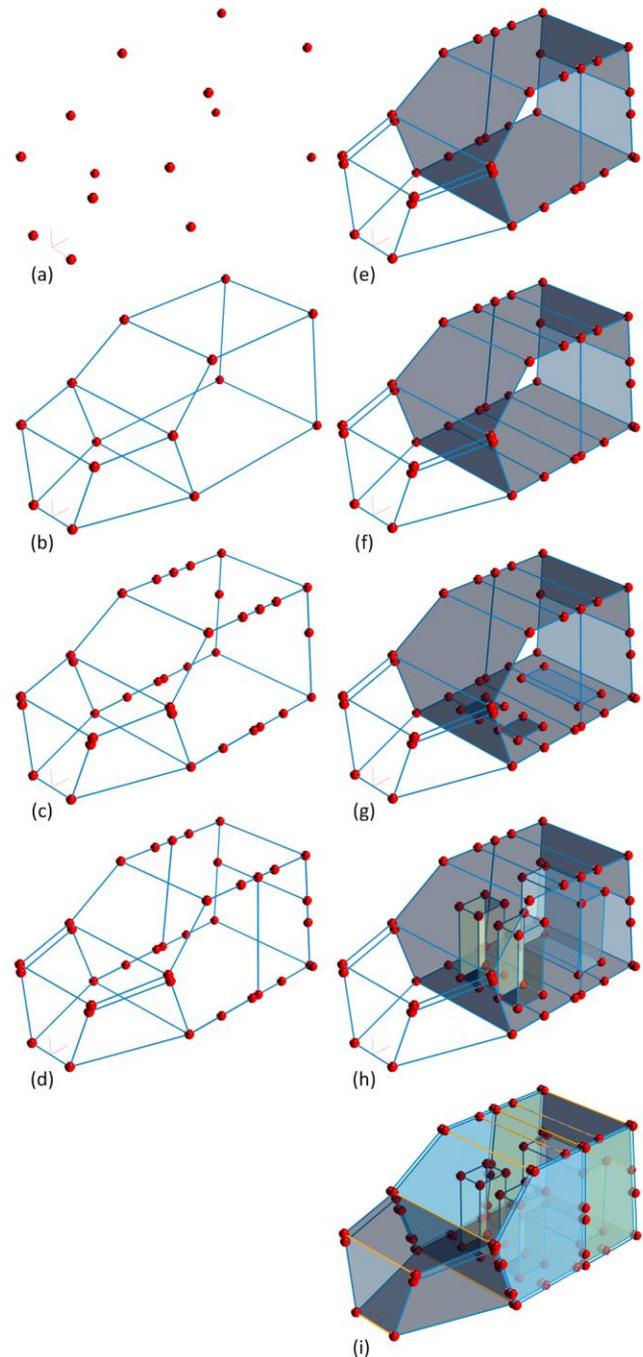

Figure 11. Pickup truck 4 door cab scaffold modeling steps defining architectural layout.

The dependency relationships between points, paths, surfaces, and enclosures of the scaffolding model are easily stored in a directed graph structure. Adding a geometric primitive to the directed graph only requires specification of the parent primitives for the primitive to be added to the model as outlined in the steps a-h. Independent points have a model root

as their parent. Removal of a geometric primitive requires the removal of any dependent geometric primitives of the one being removed as well and can be performed via a depth first graph traversal. When the geometry of a geometric primitive is changed in some manner in the scaffold model, the underlying directed graph is used again to update the geometric primitives dependent on the modified primitive by performing a breath first search graph traversal. Thus, changes flow through the model beginning with the first modified primitive and ending with the last descendant.

Once the rudimentary geometry defining the vehicle layout is specified, the designer can begin specifying which paths in the scaffold model serve as the primary load carrying members, by marking them as beam components. Surfaces may be marked as panel components to carry in-plane loads through strain energy storage. Enclosures may account for inertial components by marking them as such. By identifying the geometric primitives as components, it is possible to determine the appropriate model connectivity based on the underlying scaffold model dependencies and place them into the graph. For instance, any independent point with more than one path marked as beam component dependent on it must have a MCJ defined at that location. Similarly, any surface marked as a panel component dependent on a path marked as beam component requires a path based component connection. Likewise, any surfaces marked as a panel that is dependent on another surface marked as a panel component requires a surface based component connection. By iterating through all (43 in total) of the dependency and component type permutations, it is possible to construct a fully connected default model using default component and connection properties that can be adjusted later in the design. If a geometric primitive is unmarked as a component, then any dependent component connections are removed from the underlying directed graph.
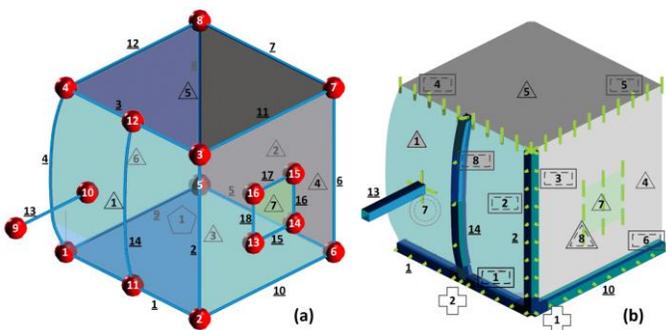

Figure 12. Representative assembly architecture including a) the underlying scaffolding primitives and b) components with connectivity. Labeling of the items correlates to Figure 13 where underline numbers represent a path or beam

The underlying directed graph data structure for the scaffolding model can become quite complex for a truck cab architecture. Thus, a simpler example of a box with front, side and top panel components and four beam components, one of which extends off of the front of the box, is included as shown in Figure 12 to

illustrate the scaffold model dependency and most common component connectivity types stored in the directed graph, Figure 13. The geometric primitives are labeled in both figures for correlation and are represented in the graph as nodes with circular, rectangular, triangular, and pentagonal shapes corresponding to points, paths, surfaces, and enclosures, respectively. Any geometric primitives in the graph marked as a component are identified by shading the shape grey. Component connections determined from the graph structure descend from only those nodes marked as components and do not have any further descendants (dependencies). They are identified as shapes with an inner dashed and outer solid border using pluses, circles, rectangles, and triangles to represent the MCJ, point based, path based, and surface based component connections, respectively.
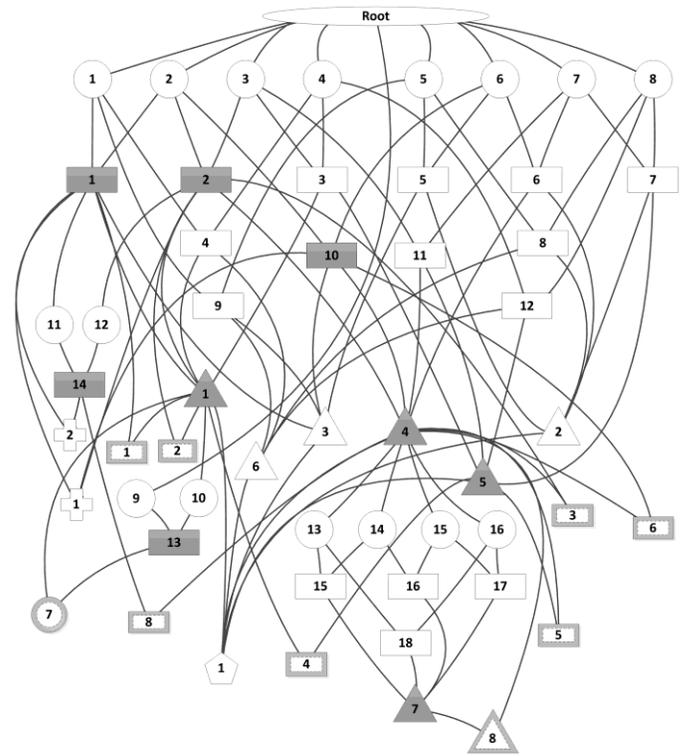

Figure 13. Directed graph depicting the dependency of geometric primitives, components, and connections for the assembly architecture in Figure 12. Edge directions point from top nodes to bottom nodes.

To further illustrate the primitive dependencies, by moving the first point in space, the first, fourth, and ninth paths' path length and orientation change, Figure 14. The alteration of the first path results in the eleventh point moving and thus the fourteenth path changes length as well. The first, third, and sixth surfaces' change shape as a result of the boundary path changes just mentioned and the enclosure geometry alters to follow these surface modifications. The alteration to the path geometry of the first beam component mandates geometric changes to the first

and second MCJs, the first, second, fourth, and eight path based connections, and the seventh point based connection.
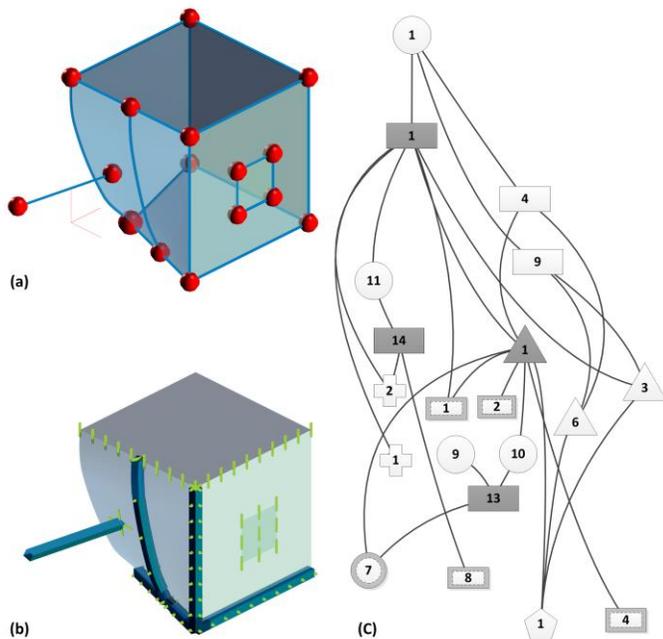

Figure 14. First independent point modification of the assembly in Figure 12 changes to the a) scaffold model, b) assembly model based on the scaffolding, and c) directed graph propogation.

Scaffolding Constraints Points within the scaffolding can be constrained to other points within the scaffold model. Available constraints on independent points include symmetry across any of the three Cartesian axes, individual common coordinate constraints, and relative coordinate constraints. Dependent point constraints include parametric symmetry constraints on a given component and common coordinate constraints across components. These constraints define relationships between points in the model that establish how a point is free to move relative to another point. The relative coordinate constraints define the position of one point relative to another so that model parameters related to the independent points can be defined and modified more intuitively. Also, by defining relative constraints, it is possible to generate custom templates for various vehicle architectures such as ladder frames, hatchbacks, sedans, station wagons, pickup truck cabs, cargo boxes, flat racks, SUVs, vans, etc., and make rapid design changes in order to quickly develop new potential design iterations for optimization of the architecture layout. These model constraints are stored in the directed graph also as edges representing the dependency of one point upon another.

Positioning and orienting the assemblies within the vehicle is performed at the vehicle level. This is the only modeling level in which a designer can see the interrelationships among the assemblies in the vehicle. By moving the assemblies at the vehicle level, clearance between assemblies can be checked

algorithmically. Assembly connectivity including the energy and power transmission paths is defined at the vehicle level as well for similar reasons. Possible energy and power transmission connections depend on the type of assemblies being connected and any existing connections for the assemblies. The physical assembly connections representing body mounts, suspension compliance elements, etc, between assemblies have no limitations on the number or types of assemblies to which a given assembly may be connected. The only condition is that all assemblies must be connected in some manner to create a contiguous vehicle in order to perform rigid body or FE analysis at the vehicle level.

Individual components and connections for an assembly are specified by opening the particular component or connection within the assembly that it is contained within. The available properties for specification depend on the type of component or connection being edited. Architecture layout operations for the individual components however, are specified at the assembly level using the scaffolding interface.

VEHICLE LAYOUT MODULE

A vehicle layout module provides an easy means for instantiating a full vehicle concept model based on a minimal set of vehicle level parameters. Vehicle parameters during instantiation include specifying the types of assemblies to include in the vehicle, suspension configurations, braked and driven wheels, powertrain categorization, and nominal sizes of the assemblies. There are five tabs, one representing each major assembly type, including frame, cab, payload, suspensions, and powertrain, Figure 15.

Options for not including a frame, cab, or payload exist, but at least one of the three types must be included in the vehicle. The suspension tab permits the designer to specify the quantity of axles, inclusion of specific brake types, dual wheels option, tire size for wheels on the axle, track width, and if the axle is driven or not. Options under the powertrain tab include conventional IC engine, electric, hybrid-electric parallel and series, and hybrid hydraulic powertrain configurations. A vehicle resembling the class type of vehicle specified by the designer is automatically generated including all necessary powertrain components and energy and power flow connections. Clearly, the designer must adjust the architecture layout, member sizes, panel gauges, powertrain component parameters, exact assembly locations, etc., but a large number of features that are easily modified through the vehicle hierarchy have been established by the vehicle layout module.
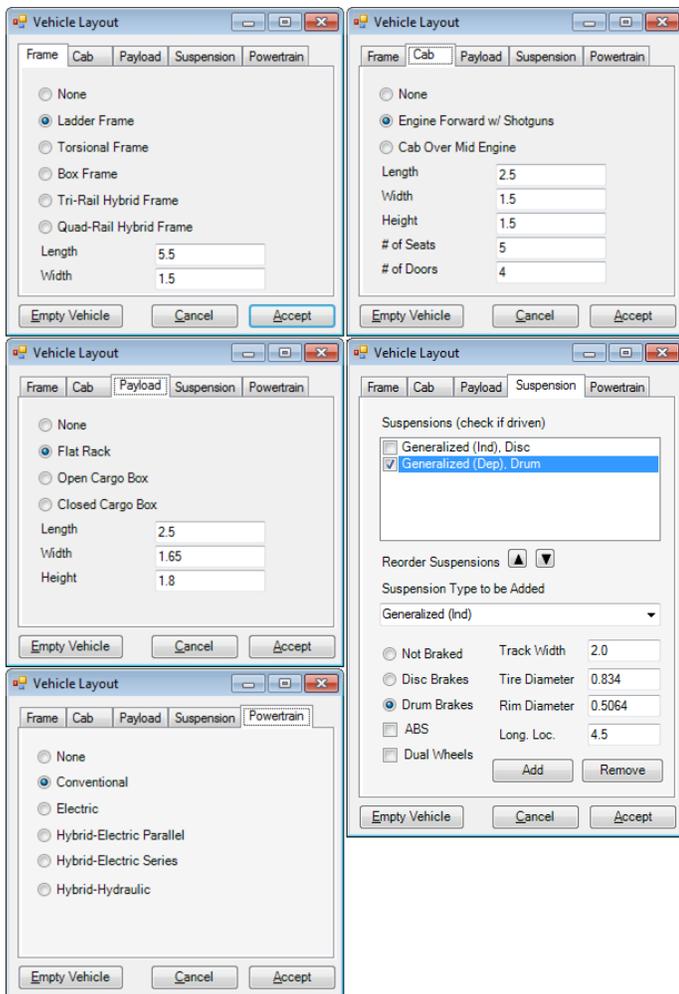
Figure 15. Vehicle architecture layout module showing assembly types and configuration parameters for vehicle initialization.

CONCLUSIONS

An interactive design interface consistent with vehicle conceptual design information and methods has been presented. The interactive interface supports both the design and analysis of vehicles during the conceptual design stage. The software provides a quick starting point for the development of many common vehicle platforms and implements a hierarchal approach that supports an iterative architecture layout optimization process necessary for addressing critical NVH issues before vehicle architecture is established.

ACKNOWLEDGMENTS

REFERENCES

[1] Natarajan, V., Chidamparam, P., Murthy, L., Gawaskar, S. and Dange, M., 1995, "Role of Concept Models in Leading Automotive Body Design," International Body Engineering Conference and Exhibition, Detroit, Michigan, USA, Oct. 31-Nov. 2.

[2] Jee, S.H. and Yi, J.C., 2000, "The Application of the Simulation Techniques to Reduce the Noise and Vibration in Vehicle Development," 2000 FISITA World Automotive Congress-Automotive Innovation for the New Millennium, Seoul, Korea, June 12-15.

[3] Schelkle, E. and Elsenhans, H.H., 2000, "Integration of Innovative CAE Tools in the Concept Stage of Vehicle Development," Numerical Analysis and Simulation in Vehicle Engineering, Würzburg, Germany, Sep. 14-15.

[4] Schelkle, E. and Elsenhans, H., 2002 "Virtual Vehicle Development in the Concept Stage – Current Status of CAE and Outlook on the Future," Proceedings for the 3rd Worldwide MSC Software Aerospace Conference & Technology Showcase.

[5] Lee, K.W., Chong, T.H. and Park, G.J., 2003, "Development of a Methodology for a Simplified Finite Element Model and Optimum Design," Computers and Structures, **81(14)**, pp. 1449-1460.

[6] Lee, K. and Nikolaidis, E., 1992, "Identification of Automotive Structures with Flexible Joints," AIAA Journal, **30(2)**, 482-89.

[7] Lee, K. and Nikolaidis, E., 1992,"A Two-Dimensional Model for Joints in Vehicle Structures," Computers and Structures, **45(4)**, pp. 775-784.

[8] Nikolaidis, E. Zhu M., 1996, "Design of Automotive Joints: Using Neural Networks and Optimization to Translate Performance Requirements to Physical Design Parameters," Computers and Structures, **60(6)**, pp.989-1001.

[9] Osborne, G., Shahhosseini A., and Prater, G., 2010, "Finite Element Concept Modelling Methodologies for Pickup Truck Boxes," Int. J. of Heavy Vehicle Systems, **17(1),** pp. 1-17.

[10] Enoki, S., Aoyama, E., Hirogaki, T., Ando, H., Kono, D. and Katayama, T., 2003, "Simple Finite Element Model with Beam Elements for :Panel Part of Spot-Welded Frame-Panel Structure," Nippon Kikai Gakkai Ronbunshu, C Hen/Trans. of the Japan Society of Mechanical Engineers, **69(3)**, pp. 805-11.

[11] Prater, G., Shahhosseini, A.M., State, M., Furman, V. and Azzouz, M., 2002, "Use of FEA Concept Models to Develop Light-Truck Cab Architectures with Reduced Weight and Enhanced NVH Characteristics," SAE Technical Paper No. 2002-01-0369.

[12] Hou, W., Zhang, H., Chi, R., and Hu, P., 2009, "Development of an Intelligent CAE System for Auto-Body Concept Design," Int. J. of Automotive Technology, 10(2), pp. 175-180.

[13] Kojima, Y., 2000 "Mechanical CAE in Automotive Design. R&D," Review of Toyota CRDL, **35**, pp. 1-10.

[14] Shahhosseini, A., Prater, G., and Osborne, G., 2010, "Major Compliance Joint Modelling Survey for Automotive Body Structures," Int. J. of Vehicle Systems Modelling & Testing, **5(1),** pp. 1-17.