2003-01-1541

Building 3D SEA Models from Templates – New Developments

Denis Blanchet and Andrew Cunningham
Vibro-Acoustic Sciences Inc.

Copyright © 2003 SAE International

ABSTRACT

Automobile sound package design requires that a Statistical Energy Analysis (SEA) model be built during initial stages of any vehicle program. This allows design changes, noise path analysis and optimization of the sound package to be performed before any program design freeze. The 3D model building process becomes a critical element since it involves many weeks of work before the model is ready for sound package definition and analysis. This paper introduces new developments in building 3D SEA models using templates. A new set of tools has been developed to further automate the SEA model building process. These tools should enable the user to develop a full vehicle SEA model within a few days. The productivity improvement gained by reducing model building time will allow for a more effective use of SEA models in the vehicle development cycle.

INTRODUCTION

With the continuing trend of automobile manufacturers to develop replacement vehicles for existing platforms and to produce new market segment vehicles, CAE simulation requirements are overloaded. This trend also affects the tier 1 and 2 suppliers as they are tasked with supplying proposals for all the new vehicles without any guarantees of either the vehicle model being targeted for production or of their company receiving the contract to supply product. The value of the CAE simulation tools is in being able to provide accurate predictions of performance with minimal cost. In the sound package design arena SEA has recently demonstrated its capability of meeting this value via the implementation of model building tools to minimize the effort required to build models of an existing vehicle architecture. The enhancements of this technology, presented in this paper, demonstrate that it is now possible to substantially minimize model creation time for classical as well as for hybrid vehicle architectures.

The template approach has proven to be an efficient method to improve efficiency in building SEA models [1]. Once a template is morphed to its new geometry, a user might want to add or remove some details on specific sub-assembly of a model, exchange a sub-assembly from one model to the other or get the TL from a user-defined group of subsystems. The new tools presented in this paper allow the user to:

- Split, merge structural subsystems with the process being extended to connected cavities
- Import a sub-assembly (floor, roof, dash…) into a full vehicle model. All connections being handled automatically
- Calculate the TL of multiple structural subsystems from within a full vehicle model
- Create a Virtual Transmission Loss (VTL) suite automatically

These new tools are intended to increase efficiency in model building using AutoSEA2. They can be used with the template approach to harvest the full potential of the method. In this paper the current template approach is briefly described and the new tools are then described in details.

OVERVIEW OF CURRENT TEMPLATE APPROACH

The 3D SEA models created with AutoSEA2 TM software are formulated from a family of “subsystems” which are defined by 3D nodes with x,y,z, coordinates [2]. The boundary shape of each subsystem is quite general, being defined by any number of 3D node points. Each subsystem is related to a set of 3D node points from which AutoSEA2 automatically calculates geometric properties such as length, perimeter, area, and volume. One can take advantage of the fact that all subsystems will see their 3D shapes and geometric properties updated automatically if one of their referenced 3D node points is relocated. The same is true for SEA properties. In fact, any AutoSEA2 TM model can be transformed into a fully parametric SEA model by automatically controlling the location of a set of nodes relative to a set of reference nodes. The “Template Modeler Method” (TMM) steps can be summarized as follows (Figure 1):

1. Import a template of generic geometry (or any SEA model)
2. Automatically create a stick model with reference nodes
3. Move reference nodes to fit new geometry
4. Automatically morph to new geometry
5. Mirror to full vehicle model (if needed)
6. Move nodes of non-symmetric subsystems

TMM uses a template to determine the geometric relationships between reference and non-reference nodes. A template can be any SEA model as long as the subsystem partitioning suits the system to be modeled. For instance, the SEA model of a 4-door sedan can easily be morphed from a small entry-level front wheel drive vehicle to a large luxury vehicle. Built-in expertise and best practices should be included in the template in order to pass this knowledge from one model to the other.

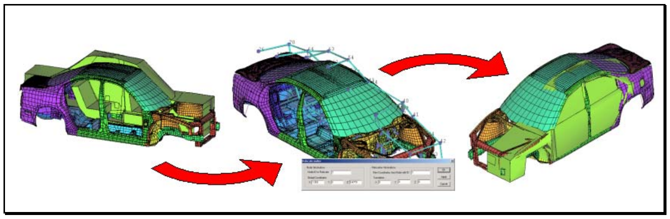

Figure 1: TMM – Load template, create stick model, move selected nodes, morph template.

Different methods can be used to morph a 3D model to a new geometry. Methods based on PID's (Property IDentification numbers) defined in an FEA model have shown promising results but are limited in application. Methods based on manually defined relationships between reference and non-reference nodes are time consuming and tedious to implement even if they provide a significant timesaving compared to building a model from scratch [3]. An alternative TMM approach automatically extracts all geometric relationships between reference and non-reference nodes from the template. The only action the user has to take is to create an input file containing a list of relocated reference nodes and their (x,y,z) coordinates in ASCII text. Once a suitable template is chosen, a stick model is automatically created to ease the process of relocation of a selected number of reference nodes. By default, all point junctions in the template are considered as reference nodes and are included in the stick model. A stick is generated at the location of each line junction in the template, joining all reference nodes. Once the stick model is created, it is loaded with the final CAD/FEA geometry information and reference nodes are relocated manually. Useful node relocation tools have been developed to ease this process. Any reference node in the stick model can be deleted and will be considered in future steps as a non-reference node. This is useful to limit the number of nodes to manually relocate. Reference nodes ID and their new coordinates are then saved in a text file for future use. The next step in the TMM process is the relocation of the non-reference nodes. This is automatically accomplished using an algorithm that analyzes the template's initial geometry to determine the linear relationships between all non-reference nodes and the reference nodes. The Template Modeler then loads the reference node text file and calculates the final location of all non-reference nodes based on the initially stored linear relationships and final location of reference nodes. TMM then calculates the final location of all non-reference nodes based on the linear relationships and location of reference nodes. All reference and non-reference nodes are then saved to a text file and can be imported into the template to complete the morphing operation. The last step in the parametric modeling is to update the full vehicle model with trim information and include any unique components for the vehicle program such as the front-of-dash, instrument panel, cargo area, etc. This step will be the most time consuming in the TMM process.

OVERVIEW OF NEW FEATURES

The new features introduced here are used in conjunctions with the TemplateModeler method and are usually used once the template is morphed into the final geometry. In the course of a vehicle design program, an SEA model of different level of details is often required. At the beginning of the design process, general acoustic performance of the vehicle are needed for the full vehicle system to establish global performance or to set target for different regions in the vehicle.

Later on in the design process, the SEA model will have to address component level issues. At this stage, a more detailed model is needed of the according component, even though details are not yet defined for all components. The component level model should be isolated from the vehicle to study TL performance and validate the component level SEA model. It should also be integrated to the full system model in order to assess the effect of a parameter change in the studied component on the overall acoustic performance of the car. Finally, closer to the last stages of the vehicle design program, the full vehicle model should include all available details.

This paper presents new features designed to allow added flexibility in 3D model manipulation. It is now possible to export the subsystems of a section of a model into a Virtual Transmission Loss (VTL) suite, add or subtract details to the exported section and import the newly modified section back into the full vehicle model.

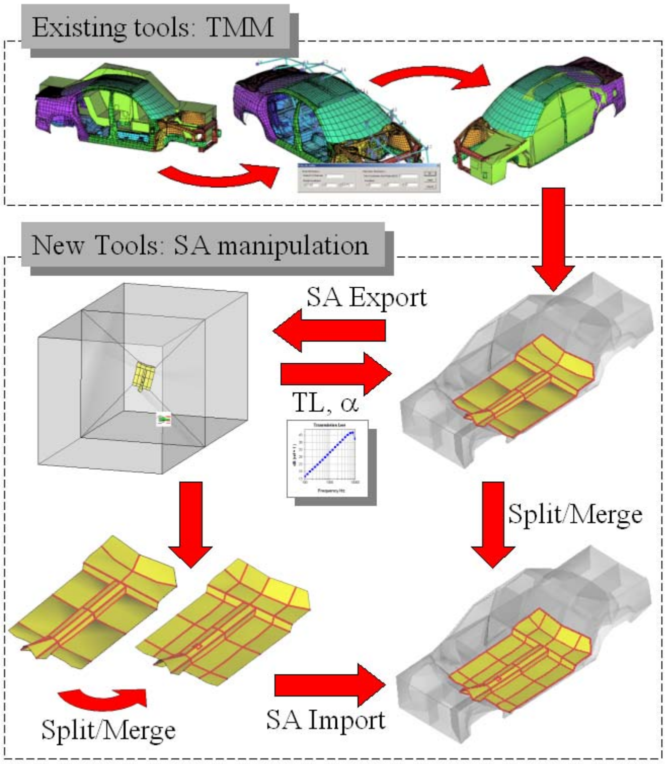

Figure 2 - Overview of new features relationship with TemplateModeler

This is done with the use of a simple naming convention and sophisticated algorithms that perform all necessary recreation of acoustic cavities and connections in order to end up with a valid AutoSEA2 model.

DEFINITION

Node – a geometric point in space created either manually in AutoSEA2 or imported from an external CAD file or finite element model file for use in defining geometry and subsystem connectivity. A node has an ID number and XYZ coordinates

Subsystem (SS) – a structural or acoustic element of the system, defined by a physical region based on points and properties based on the AutoSEA2 idealized subsystem type

Template – A SEA model of generic geometry that is used as a starting point to build a vehicle SEA model

Sub-Assembly (SA) – A group of structural subsystems (shells) connected together to form a component of the car (ex: floor, roof…). In this project, a sub-assembly is defined by a naming convention of the structural SS defined in the SA text file

Sub-Assembly external edges – The external boundaries around the sub-assembly that defines the node Id's that will connect the sub-assembly to the structural subsystem of the full vehicle template. (see Figure 3)

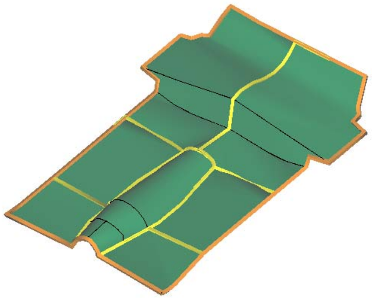

Figure 3: Orange: Sub-Assembly edges, Yellow: internal cavity edges, Black: SS edges

Sub-Assembly internal edges – The internal boundaries of the sub-assembly that defines the node Id's that connect the sub-assembly structural subsystems to the cavities of the full vehicle template. An example of the floor sub-assembly internal and external edges as well as SS edges is presented in the following figure:

Sub-Assembly (SA) text file – A text file describing the sub-assembly name in the first column and its subsystem naming convention keyword in the second column. SA names and subsystem naming convention are user-defined and used by the script GUI dialogs.

SUB-ASSEMBLY IMPORT SCRIPT

This script is designed to allow the user to replace an existing sub-assembly (SA) such as the floor of a vehicle (Figure 3) with a coarser or more detailed SA. The ***"SA import"*** script only imports the structural subsystems of the file that contains the new SA. This is useful when the new SA subsystems are included in a VTL suite since it can be directly imported in a full vehicle model without any pre-formatting of the VTL file. In the import process, all connections between cavities in the initial model and the new imported SA structural subsystems are handled automatically. No extra work is needed to manually connect the initial cavities with the newly imported subsystems. This requires a sophisticated algorithm to be written. The following presents the approach adapted to perform this task.

Firstly, all the structural subsystems in the initial model that belong to a particular sub-assembly are found based upon a naming convention (e.g. all shells that belong to the sub-assembly "Floor" have a name starting with "SA_Floor"). These shells are then grouped according to which cavity they are connected to. Each one of these groups is a candidate for replacement by an "equivalent" set of shells in the SA being imported. Equivalent, in this case, means that a set of shells matches another set of shells if each set has an equivalent list of boundary nodes IDs. The boundary of a set of shells is found by constructing a "directed graph" data structure where the nodes of the graph are the shell 3D nodes IDs, and the edges of the graph are the edges of the shells. Finding the boundary of a set of shells corresponds to finding the "free" edges of the graph, that is, those edges that are shared by only one subsystem (holes are not allowed). From this list of free edges, one can then construct a list of boundary node IDs. The objective is then to find a list of shells in the new SA that have the same boundary as the initial SA shells identified above. The algorithm builds a undirected graph of connectivity of the subassembly shells. In this graph the nodes are the shells themselves, and the edges indicate that two shells are connected. All the SA shells that share an edge with the boundary of interest are found and divided into two lists. One list consist of shells that make up the positive side of the boundary, the other list the negative side. Both sides need to be considered, as the boundary of the initial shells is not necessarily consistent with the winding of the imported sub-assembly shells (Figure 4).

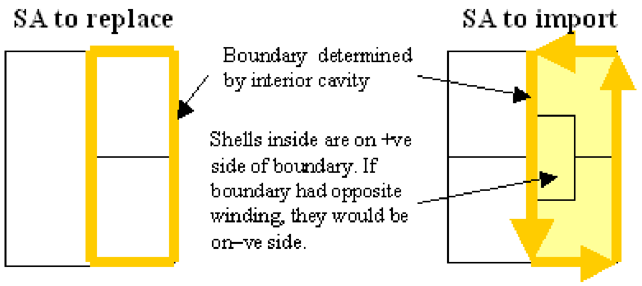
SA to replace
SA to import
Boundary determined by interior cavity
Shells inside are on +ve side of boundary. If boundary had opposite winding, they would be on–ve side.

Figure 4: Handling automatic connection between initial model and imported SA

Firstly, the algorithm tries the shells on the positive side by removing the negative shells from the subassembly (SA) connectivity graph. This has the effect of breaking the graph into two sub-graphs. By performing a BFS (breadth first search) of the sub-graph containing the positive shells, the algorithm attempts to match all the shells found in this search. If no matches are found, the algorithm tries the negative side shells.Once all relationships are determined, the "SA import script" recreates all appropriate cavities in order to benefit from the *autoconnect* feature of AutoSEA2 that automatically create all junctions between newly imported SA and all appropriate subsystems in the initial model.

SPLIT/MERGE SUBSYSTEMS SCRIPT

Manual splitting of a subsystem implies that the user click on a number on nodes on the subsystem to split in order to create the first split plate. The same procedure has to be followed with the second split plate. Once the two plates are created, the initial plate is deleted. Then, any cavity connected to the initial plate has to be recreated using the new split plates in order for them to share the same face. This is an imperative step since AutoSEA2 requires that plates and cavity share the same face in order to automatically create an area junction connecting the two subsystems. The manual splitting of a SS is a tedious operation.

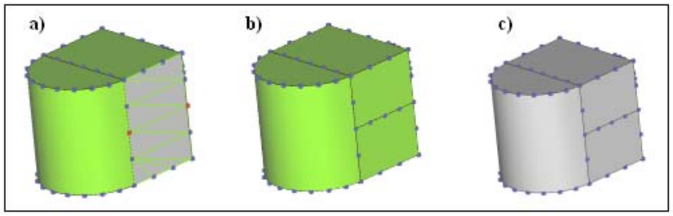

Figure 5: a) Select SS to split, and nodes where to split. Specify number of nodes on shared edge. b) SS is split, at specified nodes, with specified number of nodes on shared edge (3 here). c) Settings of all connected cavities are copied and cavities are automatically recreated to share new SS faces.

To overcome this problem, the ***"Split/Merge SS"*** script has been developed. This new script allows the user to automatically split a flat plate or a singly-curved shell into two new subsystems. It also allows for merging of two SS into one. To split a SS, the user selects the plate to split and two nodes indicating where to split the SS. The script will determine whether a flat or singly-curved shell is selected and prompt the user for a number of nodes to add to the new edge to be created if necessary. Figure 5 shows an example of the splitting of a flat plate.

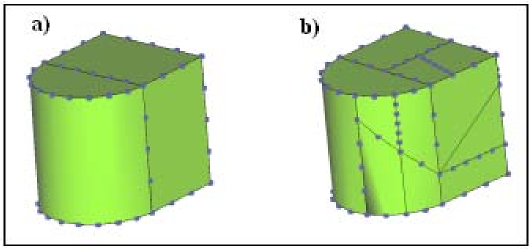

Figure 6: a) Initial model b) Split/Merge SS script can be used with flat plates and singly curved shells in a wide variety of ways.

Splitting of a singly-curved shell follow the same creation restriction as the manual operation. There has to be a equal number of nodes on the curve edge of each singly-curved shells with three nodes being the minimum number of nodes on these edges. This restriction implies that when splitting a singly-curved shell along the curved edge, the script will determine automatically how many nodes to add to the newly created edge.

The ***"Split/Merge SS"*** script also allows the user to merge flat or singly-curved shells as long as the selected SS are of the same type. Merging also automatically recreate any connected cavity as needed.

VIRTUAL TRANSMISSION LOSS (VTL) SCRIPT

The ***"Virtual Transmission Loss (VTL)"*** script allows the user to select a number of contiguous subsystems (SS) in a model and computes the resulting composite TL of the selection using eq. 1. The later corresponds to the effective TL result graph calculation in AutoSEA2 [4].

$$TL = 10 \log_{10}\left(\frac{A\omega}{8\pi^2 c_1^2 n_1 \eta_2}\left(\frac{E_1}{E_2}-\frac{n_1}{n_2}\right)\right) \quad (1)$$

where A is the total surface area of structural subsystems, ω is the center band angular frequency, c is the speed of sound, n is modal density, η is damping loss factor, E is energy level and subscript *1* refers to source room and *2* for receiver room. The TL spectrum is then stored in the current database.The ***"VTL"*** script also automatically generates a VTL suite in a separate file for future use. A VTL suite is defined as an SEA model containing a source and receiver cavities and a number of contiguous structural subsystems connected to these cavities. It also includes all necessary junctions and a power source connected to the source cavity. To generate the VTL suite, the edges of the selected subsystems are found using the algorithm described in the ***"SA import"*** script section. A plane of best fit is found for this list of boundary nodes. The "best" four corners of the boundary are found by using a 3D iterative repulsion algorithm. From the plane of best fit and "best" four corners, the bounding cavity corners and faces can be constructed, and projected to form the receiver and source cavities (Figure 7). The ***"VTL"*** script can generate a wide variety of subsystem collection such as a complete floor, dash or even double-wall systems such as a door (Figure 8).

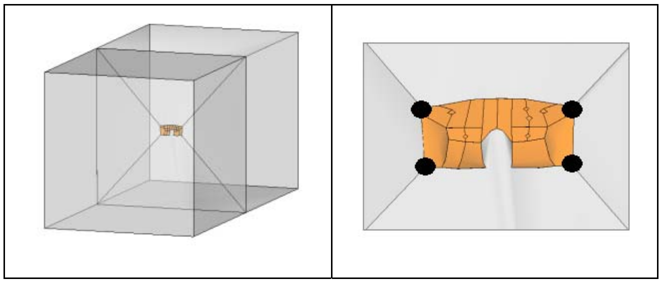

Figure 7: a) VTL suite automatically created. b) Four "best" corners of the selected subsystems.

The resulting VTL suite can then be used to refine a component SEA model, and perform parametric study using the TL as the objective function.

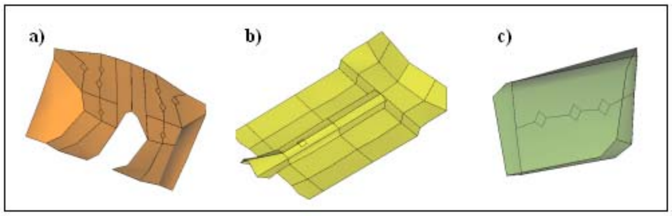

Figure 8: VTL script can automatically create VTL suite of complete component such as: a) dash, b) floor, c) double-wall systems such as a door

The is done by the use of the ***"Split/Merge SS"*** script. If needed, the resulting sub-assembly model can be imported back in the initial full vehicle model using the ***"SA import"*** script.

The VTL suite is built according to classical TL definition. The source room modal density depends on volume only since the surface area and perimeter are set to 0. The receiver room absorption is set to the ratio of structural elements surface area over the total receiver room surface area (A_{panels} / $A_{receiver\ cavity}$). This forces the NR to be equal to the TL as can be derived from eq.2 [5].

$$TL = 10 \log_{10}\left(\frac{W_{inc}}{W_{trans}}\right) = 10 \log_{10}\left(\frac{P^2_{source}}{P^2_{receiver}}\right) + 10 \log_{10}\left(\frac{A_{panel}}{\overline{\alpha}_{room}}\right) \quad (2)$$

where W is acoustic power, P^2 is mean square pressure, A is total surface area and α is the average absorption in the receiver room. The transfer function between the source and receiver room then yields the TL of the structural elements. The later can easily be graphed in AutoSEA2. The VTL suite is therefore an efficient way to assess TL performance of a complete sub-assembly.

CONCLUSION

New developments in the process of building SEA models were successfully developed. The new tools can be used in conjunction with the TempleteModeler Method of building fast and standardized model. These new tools add flexibility and efficiency in the refinements of SEA models. They also allow the exchange of sub-assemblies from one model to the other.

REFERENCES

1. R. Unglenieks and D. Blanchet, "Analysis of Interior Noise of Vehicles Using An Automated SEA Model Building Tool - A Case Study", Proceedings of Internoise 2002, Dearborn
2. P. Bremner, T. Burton, and A. Cunningham, "AutoSEA2 - A new design evaluation tool for noise and vibration engineering", Proc. 1999 ASME Design Engineering Technology Conferences, September 12-15, 1999, Las Vegas.
3. D. Blanchet, A new automated SEA tool for the analysis of interior noise of automobiles, Proceedings of Internoise 2001, 2441-2445, 2001
4. Shorter, P., *Calculating transmission loss in an SEA model*, Technical Memorandum no: VAS-1066-01, Vibro-Acoustic Sciences, Dec 2000
5. AutoSEA Theory and Quality Assurance Manual, version:1.4, rev 0.