Immersion Product Configurator for Conceptual Design

Ruqin Zhang  
_Iowa State University_

Christian Noon  
_Iowa State University_

Eliot H. Winer  
_Iowa State University_, ewiner@iastate.edu

James H. Oliver  
_Iowa State University_, oliver@iastate.edu

Brian Gilmore  
_John Deere Moline Technology_

See next page for additional authors

Follow this and additional works at: https://lib.dr.iastate.edu/me_conf

Part of the Computer-Aided Engineering and Design Commons, and the Computer Sciences Commons

Recommended Citation

https://lib.dr.iastate.edu/me_conf/107

This Conference Proceeding is brought to you for free and open access by the Mechanical Engineering at Iowa State University Digital Repository. It has been accepted for inclusion in Mechanical Engineering Conference Presentations, Papers, and Proceedings by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Immersive Product Configurator for Conceptual Design

Abstract
Currently, new product concepts are evaluated by developing detailed virtual part and assembly models with traditional Computer Aided Design (CAD) tools followed by appropriate analyses (e.g., finite element analysis, computational fluid dynamics, etc.). The creation of these models and analyses are tremendously time consuming. If a number of different conceptual configurations have been determined, it may not be possible to model and analyze each of them. Thus, promising concepts might be eliminated based solely on insufficient time to assess them. In addition, the virtual models and analyses performed are usually of much higher detail and accuracy than what is needed for such early assessment. By eliminating the time-consuming complexity of a CAD environment and incorporating qualitative assessment tools, engineers could spend more time evaluating additional concepts, which were previously abandoned due to time constraints. In this paper, a software framework, the Advanced Systems Design Suite (ASDS), for creating and evaluating conceptual design configurations in an immersive virtual reality environment is presented. The ASDS allows design concepts to be quickly modeled, analyzed, and visualized. It incorporates a PC user interface with an immersive virtual reality environment to ease the creation and assessment of conceptual design prototypes. The development of the modeling and assessment tools are presented along with a test case to demonstrate the usability and effectiveness of the framework.

Keywords
Virtual Reality Applications Center

Disciplines
Computer-Aided Engineering and Design | Computer Sciences

Comments

Authors
Ruqin Zhang, Christian Noon, Eliot H. Winer, James H. Oliver, Brian Gilmore, and Jerry Duncan

This conference proceeding is available at Iowa State University Digital Repository: https://lib.dr.iastate.edu/me_conf/107
IMMERSIVE PRODUCT CONFIGURATOR FOR CONCEPTUAL DESIGN

Ruqin Zhang
Research Assistant
Virtual Reality Application Center
Dept. of Mechanical Engineering
Iowa State University
Ames, Iowa 50011-2274
ruzhang@iastate.edu

Christian Noon
Research Assistant
Virtual Reality Application Center
Dept. of Mechanical Engineering
Iowa State University
Ames, Iowa 50011-2274
cnoon15@iastate.edu

Eliot Winer
Assistant Professor
Virtual Reality Application Center
Dept. of Mechanical Engineering
Iowa State University
Ames, Iowa 50011-2274
ewiner@iastate.edu

James H. Oliver
Professor
Virtual Reality Application Center
Dept. of Mechanical Engineering
Iowa State University
Ames, Iowa 50011-2274
oliver@iastate.edu

Brian Gilmore
Manager
Advanced Systems Engineering
John Deere Moline Technology
Innovation Center
Moline, IL 61265
gilmorebrianj@johndeere.com

Jerry Duncan
Manager
Collaborative Science
John Deere Moline Technology
Innovation Center
Moline, IL 61265
duncanjerryr@johndeere.com

ABSTRACT

Currently, new product concepts are evaluated by developing detailed virtual part and assembly models with traditional Computer Aided Design (CAD) tools followed by appropriate analyses (e.g., finite element analysis, computational fluid dynamics, etc.). The creation of these models and analyses are tremendously time consuming. If a number of different conceptual configurations have been determined, it may not be possible to model and analyze each of them. Thus, promising concepts might be eliminated based solely on insufficient time to assess them. In addition, the virtual models and analyses performed are usually of much higher detail and accuracy than what is needed for such early assessment. By eliminating the time-consuming complexity of a CAD environment and incorporating qualitative assessment tools, engineers could spend more time evaluating additional concepts, which were previously abandoned due to time constraints. In this paper, a software framework, the Advanced Systems Design Suite (ASDS), for creating and evaluating conceptual design configurations in an immersive virtual reality environment is presented. The ASDS allows design concepts to be quickly modeled, analyzed, and visualized. It incorporates a PC user interface with an immersive virtual reality environment to ease the creation and assessment of conceptual design prototypes. The development of the modeling and assessment tools are presented along with a test case to demonstrate the usability and effectiveness of the framework.

Keywords: Product configuration; Conceptual design; Virtual reality

1. INTRODUCTION

Product design is an information intensive engineering process of decision-making. It is estimated that as much as 75% of the cost of a product is spent during the product design phase including manufacturing and maintenance (1). Companies are increasingly using digital prototypes such as CAD or free form models from programs such as Solidworks, ProEngineer, Maya, 3D MAX, etc. (2) rather than manufacturing expensive physical models earlier in the product development process. Product design usually starts with the definition of a design problem, followed by a sequence of approaches to find an optimal solution and ends with a detailed description of the product. A design process can be divided into several phases. The first is collecting and defining design specifications about the product such as performance, quality, and safety. The second is concept generation where rough design concepts are proposed to meet the design specifications. Next is detailed design, where all
design specifics such as part dimensions, material specification and assembly arrangements, etc., are finalized. These 3D product models form the basis for detailed performance analysis, manufacturing planning and all other product life-cycle activities such as production and maintenance. Many computer tools have been developed to assist design and analysis at the detailed stage of design (3, 4), whereas concept generation and selection are still mostly dependent upon experience of engineers and use of tools not made to handle the requirements of concept generation.

As mechanical systems and products continue to be developed, and become increasingly more complex, the early stages of a design process—where decisions can be made with minimal cost and time impact—become more critical to the success of the resulting product. Given well-defined design requirements, it is challenging to generate and select a concept that effectively satisfies all of the requirements. Conceptual design can have significant impacts on the downstream design and manufacturing process (5). Early design stages typically include engineers identifying the needs and wants of a particular project and producing a concept pool using various creative methods such as brainstorming (6). Engineers produce as many different concepts as possible in order to have a wide variety of possibilities to evaluate at the next level of design. Depending on the project, concept generation could produce anywhere from tens to hundreds of possible concepts.

Once the pool of concepts has been established, engineers must reduce the list to a manageable number (two to three) to proceed to detailed design. Currently, there are limited tools to aid in this process. The most prevalent method is to model concepts using detailed design tools such as CAD software. However, due to the specificity needed to create a solid model, considerable time and resources are consumed producing these 3D concept models simply to assess the rough measure of feasibility needed for evaluation of a concept. Due to the complexity and information needed by detailed design tools, an adequate evaluation of every conceptual configuration of a new design cannot be evaluated, it is simply too time consuming and costly. Some current CAD programs on the market today have been “lightened” in order to produce a product less complex and simple to use, such as Pro/CONCEPT (7) and CATIA Imagine and Shape (8). However, these interfaces are still very complex offering many options and features. Thus, without extensive training, quick concept generation and assessment is still not a viable option.

So, fast conceptual modeling and assessment is currently not possible for most complex product design processes.

A description of a concept design can be decomposed into various aspects including function, behavior and structure (9). To generate and select the feasible solutions, it is necessary to determine the correlations and interactions among these aspects. There are two inherent difficulties in conceptual design activities: 1) modeling these interactions and 2) reasoning to generate and select solutions (9). To overcome these modeling and reasoning problems, some design related techniques or methodologies have been developed during past years. To find modeling representations of a new design, Rinderle et al. (10, 11) formalized design by utilizing a graph-based language to represent the specification of design and behavior of the components. To model the geometric information, Lin et al. (12) made use of variational geometry. And based on this work, Light et al. (13) extended to allow modification of geometric models by using variational geometry. While Keirouz et al. (14) proposed an integration of different representation schemes including parametric, geometry, features and variational modeling. Using artificial intelligence techniques, Rao and Lu (15) applied machine learning to engineering design to study bi-directional models that can provide design synthesis support and reduce design iterations. Arpaia et al. (16) employed both abductive and deductive reasoning techniques in the design of a measurement system.

In the field of design research, researchers developed structures and methodologies to provide clear processes for designers to generate design concepts such as axiomatic design (17) and decision-based design (18). While the well-known quality function deployment (QFD) (19) makes use of matrices to transform information of the customer need to design variables and product characteristics. Based on Keeney’s Value Focused Thinking (20, 21), Jin et al. take a value-based design (VBD) (22-25) approach in conceptual design, which utilize design value to specify what is important to the designer. The design value is defined as a group of structured design objectives and a design objective driven approach is proposed to assist design concept generation.

Concept selection methods exist in order to help engineers rank the population of concepts (estimating technical difficulty, Pugh concept selection charts, numerical concept scoring, etc.) (6). These methods have proven effective but are simply a ranking system of engineers’ opinions on each concept’s ability to meet particular criteria of the design proposal. In-depth modeling and analysis (factual hands-on information) does not play a role in these elimination sessions.

The research presented in this paper builds on these ideas to provide designers with a new collaborative tool to facilitate concept generation and assessment. The objectives of the software framework, referred to as the Advanced System Design Suite (ASDS), are to: 1) enable an engineer to quickly build a 3D model of a proposed design, 2) assess a concept with qualitative analysis and quantitative data, and 3) visualize the results on both desktop and immersive virtual reality (VR) systems. The environment enables fast geometry creation by simplifying or eliminating the inputs and interfaces that CAD systems typically require, but are unnecessary at the conceptual design phase. Another objective is to provide fast computation of general physical properties (e.g., center of gravity), and qualitative performance measures. These approximations allow engineers to accelerate assessment of new concepts, and the immersive virtual environment facilitates collaborative problem solving.

With the VR-based ASDS system, a group of engineers can create and assess multiple concept ideas in real-time. For example, the process could start by selecting from several base component geometries (e.g., chassis designs). Then, through a unique, intuitive 3D modeling system, features can be added or taken away to produce a new design concept. Typical modifications range from relatively small parameter changes, such as increasing the length of the frame by 10%, to large-scale change such as adding a third axle to a two-axle vehicle.
Following the model creation some basic properties will be computed such as vehicle weight and center of gravity. Additional output will provide information on other vehicle performance measures including load distribution and tipping moment. The engineers then have information from which to base further decisions. These decisions might include whether to proceed with this concept to a more detailed analysis or to investigate other conceptual configurations. The 3D model and assessment output will also foster new ideas to the current concept, ideas that would have been previously overlooked. Multiple iterations of conceptual designs can help design teams develop a specific list of requirements and ultimately a final direction for the team to pursue into the next phase of the design process.

The following section of the paper presents background explaining in detail the different tools implemented into the ASDS system. Next, a methodology section focuses on system architecture, interface design, and assessment tools. Then a test case is presented followed by conclusions and future work.

2. BACKGROUND

VR technology is gaining increasing utility for a variety of applications in product development (26). With real-time interactive graphics, stereoscopic display, and user tracking, VR can be particularly useful for applications in which one-to-one (e.g., human) scale is important or when the assessment of complex geometric relationships are required. Haptic interfaces have also been employed for assisting conceptual design (27).

The ASDS application consists of a client interface on a tablet, laptop, or desktop PC and a VR viewer. The ASDS employs VR Juggler (28) as the VR platform with OpenSceneGraph to manipulate the graphical scene (29). The client interface is built on a scene modeling and manipulation software package called OSGedit (30). All of these foundational software tools are available via open source licensing.

2.1. VR Juggler

VR Juggler is a platform of software technologies and tools necessary for virtual reality application development. It provides developers with a suite of application programming interfaces that abstract and simplify all interface aspects of application programming, including display surfaces, object tracking, selection and navigation, graphics rendering engines, and graphical user interfaces. Based on C++, VR Juggler achieves its goals with a growing set of generic programming tools. The suite provides a complete VR system abstraction in reusable, cross-platform modular components. Each component is decoupled from the others so that the application includes only what it needs. An application written with VR Juggler is essentially independent of devices, computer platforms and VR systems. This allows a user to run an application on almost any VR system. VR Juggler is scalable from simple desktop systems like PCs to complex multi-screen systems running on high-end workstations and computer clusters. The flexibility of VR Juggler allows applications to execute in many VR system configurations including desktop VR, HMD, CAVE-like devices, and Powerwall™-like devices. The cross-platform portability ensures its application on different operating systems including IRIX, Linux, Windows, FreeBSD, Solaris, and Mac OSX.

2.2. OpenSceneGraph

The OpenSceneGraph (OSG) is an OpenGL-based, 3D graphics toolkit, used by research and commercial developers for a variety of real-time visualization applications such as simulation, games, virtual reality, modeling, etc. OSG provides an optimized object oriented framework for fine control over the scenegraph of a graphics application.

A scenegraph is an object-oriented structure that arranges the logical and spatial representation of a graphical scene. Scenegraphs are a collection of nodes in a graph or tree structure. Each node may have many children but often only a single parent, the effect of a parent is apparent to all its child nodes. So, an operation applied to a group automatically propagates its effect to all of its members. For example, associating a geometrical transformation matrix at each group level and concatenating such matrices together is an efficient and natural way to process such operations. A common feature, for instance, is the ability to group related shapes/objects into a compound object which can then be moved, transformed, and selected as a single object. OSG contains functions to control all aspects of a scenegraph including how the tree is created to individual control over nodes and children. In addition, OSG provides 45 separate plug-ins for loading various 3D databases and image formats such as OpenFlight, 3ds, and Wavefront OBJ. Originally developed on IRIX, it is now available on a wide range of platforms including Linux, Windows, Solaris, and Mac OS.

2.3. OSGedit

OpenSceneGraph Editor (OSGedit) is a composer used to create and edit OSG scenegraphs. OSGedit allows a user to not only optimize object locations and properties in a virtual environment, but to organize the tree structure for object manipulations that will occur once the scenegraph is put in a specific software application. For example, names for a specific set of objects can be created in OSGedit and are then available in the graphics application using the created the scenegraph. Since OSGedit is based on OSG it can load the range of 3D models and image formats allowed by OSG. It also provides for run-time model creation of simple geometric primitives that can be added to a scenegraph.

3. METHODOLOGY

To develop an environment for rapid creation and evaluation of new design concepts, the system architecture shown in Figure 1 was developed. User interaction is done through the client interface on a tablet or laptop PC. The client interface is where all legacy parts models (loaded from the database source in the figure) are brought into ASDS. The client incorporates its own interactive 3D viewing window that controls all of the manipulation—rotation, scaling, panning, and translating—of the model. As shown in the figure, all changes done on the client are transmitted over a network connection and performed in the immersive viewer simultaneously. The immersive viewer also acquires legacy models from the same data source as the client. The client-immerse communication model: 1) eliminates the need for a
second interaction device, such as a wand or haptic device, inside the VR environment and 2) allows standard 2D user interfaces to control an immersive 3D environment.

![Diagram](image.png)

**Figure 1:** Schematic of ASDS system architecture.

### 3.1. Architecture

**Client Interface** — The client interface currently operates under the Windows OS and can be run separately from the immersive viewer. This is useful to prepare for a full collaborative conceptual design review *a priori*. In addition to using OSGEdit, the GIMP Toolkit (GTK+) (31) is used for GUI creation. GTK+ is a cross-platform application-programming interface (API), which enables common user interface features such as file open/save dialog windows and toolbars to be developed. With the use of OSGEdit and GTK+, the client interface could be ported to other operating systems.

**Network** — Network communications between the ASDS client and immersive applications are transported using a created UDP (User Datagram Protocol) socket program. UDP was chosen over other types of communication protocols such as TCP (Transmission Control Protocol) due to its speed—real-time immersive performance is extremely important for this application. The broadcast nature of the UDP protocol provided significantly better performance compared to that of TCP, particularly given the cluster configuration of CAVe immersive device for which ASDS was designed. Furthermore, UDP does not employ error checking as TCP does which can cause significant delays between the client and the immersive viewer.

**Immersive Viewer** — The ASDS CAVe visualization software was developed with OpenSceneGraph libraries with VR-Juggler for interface communication, stereoscopic viewing, and display-device abstraction. The immersive viewer is simply a slave to the tablet application. In multiple user environments, the immersive viewer will be the focus of the group. The client interface allows one user to manipulate the scene as the group sees fit. The purpose of having an immersive viewer for this application is to be able to visualize the concepts in a 3D VR environment to allow users a realistic sense of the various products and conceptual designs.

**Data Source** — All the legacy data (e.g., part libraries) acquired from previous designs is stored in a location accessed by both the client interface and immersive viewer. In its envisioned use the ASDS part libraries will consist of legacy product models comprised of geometric as well as physical properties of previously created products. It may also contain new parts created by other programs such as Google SketchUP which are not part of the corporate legacy data base. The physical data contained within these part libraries is used for further evaluation and analysis. This information currently consists of part weight and center of gravity information, which currently enables calculation of overall part CG and wheel loading.

### 3.2. Interface Interaction

This environment allows an engineer to easily create various concept models without the detailed, time-consuming constraints of CAD systems. Initially, engineers decide which existing models can be useful in creating and evaluating concepts for a particular design. These models are typically defeatured to a level of detail appropriate for conceptual design and, if necessary converted to a format accessible to the ASDS system. For the areas lacking model representation, ASDS has two options: first, it can quickly create primitive geometry on-the-fly to replace missing geometrical representations in new designs (these primitives include spheres, cylinders, cones, and cubes); a second option is to
create the missing geometry inside of any CAD software package. A particularly useful tool for quick 3D component modeling is Google SketchUP which produces files that can be imported directly into the ASDS scenegraph through the Wavefront OBJ file format. Once the part model libraries have been created, ASDS can be used to quickly create numerous design concept configurations.

Currently, the product hierarchy for objects and groups is created before a model is imported into the ASDS scenegraph. This hierarchy can be manipulated by the user to build a useful and sensible structure. A sample tree structure consisting of six part “groups” is shown in Figure 2. The six groups located below BODY also contain individual components of the product (i.e. the individual parts). These pre-constructed tree structures allow the user to simply import the entire product all at once and not part-by-part thereby saving time and resources. Additional assemblies and new parts may then be appended to the existing structure, and the resulting hierarchy can be manipulated. Reusing existing hierarchies, and enabling overall hierarchy manipulation, allows the entire concept product model to be saved for reuse.

![Diagram of Combine_Group](attachment:combine_group_diagram.png)

**Figure 2:** Sample of tree structure for the scenegraph.

In addition to manipulating the product hierarchy itself, the user can add transforms such as translation, rotation, and scaling. Two different methods are provided for selecting objects for manipulation: 1) the user may graphically pick a part in the client view window or 2) select the object by name inside the tree structure located on the right-hand side of the user interface.

Another feature of the client interface is the option for the user to combine multiple parts into what we refer to as “functional groups.” A functional group is simply a collection of related objects/components that operate together as a whole to provide a service for the overall product. Once a functional group has been established, the group then acts as a single object and can be manipulated as one. Furthermore, the user has the option to break up functional groups, form different ones for future manipulations, or add additional groups inside of existing ones.

Through software testing, an obvious need for the user was the ability to disconnect the tablet from the immersive application. For example, the UDP socket only sends the message to change the immersive application once and there is no verification as to whether or not the immersive side received the message. Therefore, due to network connection issues, the immersive application sometimes does not receive data causing the two applications to not match up. Therefore, the immersive application may be missing parts or features that the client still contains. In order to address this issue, a "resynch" button was developed to resend the previous message if the immersive application failed to receive a message.

Finally, when creating a conceptual design configuration, precise CAD-like capabilities such as collision detection and precise mating specification are generally unnecessary and would only slow the concept generation process. The ASDS therefore allows ambiguity in the physical connection between individual parts within the scene. Calculation of the mass properties is based only on relative part positions within the scene graph. However, to provide the user with some absolute basis for size and dimension, ASDS provides a visual measuring tool to indicate dimensions. A visual ruler system allows the user to drag components to the appropriate position he/she chooses. A measurement system built into the software provides the user with the ability to access useful geometrical property information from the scenegraph. This is done by selecting a geometrical boundary inside the scenegraph. Once the boundary has been selected, the software returns the physical characteristics of that particular boundary such as radius, length, width, etc. Size parameters can alternatively be entered numerically in a text field for higher accuracy in designs that require such functionality.

### 3.3. Assessment Tools

Simply creating a 3D visual representation of a new concept design is extremely useful. However, associated assessment tools can give designers additional information to make educated decisions about concept integrity and viability. Additionally, the geometric models used for the assessment do not need to change in order to perform the calculations. All required data for computations is stored outside of the model geometry. The models are parametric and do not need to be tessellated for any assessment measures. With this goal in mind, some preliminary assessment tools have been integrated in the ASDS system. The first of these is a method to dynamically compute the center of gravity (CG) of the entire model which is represented, in both the client and the immersive viewer, by a red sphere. Since individual part weight and CG are stored as metadata within the scenegraph and their absolute spatial locations can be determined by concatenating all transforms in their path, the associated CG location of any subassembly, or the entire model, can be computed easily by summing moments about each coordinate axis. The CG calculation is triggered by the same events that cause the application to "redraw", so that as any part or subassembly in the scenegraph is manipulated the assembly CG responds in real time. To ensure assembly CG visibility, a transparency function allows the user to turn any object or
all that required development was a function representing the relationship between the percentage coefficient \(p\) and the distance ratio. In order to create this relationship, finite element analysis (FEA) test cases were performed and recorded in a manner seen in Figure 3.

![Figure 3: A diagram representing the finite element analysis scheme implemented to determine the relationship between the percentage coefficient and the distance ratio.](image)

The FEA method was devised with a circular base with four supports loaded with a force in the CG position. All four wheels were positioned symmetrically about the CG. Three of them were located the same distance from the CG position, while a fourth wheel’s distance was varied in different test cases in order to find a relationship between the fourth wheel load and the load of the other three wheels. This can be seen quite clearly in Figure 3. The distance ratio used in the analysis ranged from values of 1.25-5. Values larger than five were not feasible as the data showed the loading on the fourth wheel to be insignificant after reaching a distance ratio of approximately 4.36.

The FEA test case data was recorded in order to develop the relationship of the distance ratio to the percentage coefficient. In order to do this, for each test case, the distance ratio was plotted against the percentage of the fourth wheel to the largest wheel load. However, upon studying the data, it was unclear whether the most accurate method would be using the ratio of the fourth wheel versus the largest wheel load, the average of the three wheel loads, or the minimum of the three wheel loads. Therefore, all three methods were developed to be evaluated.

A plot of the data of the distance ratio versus the percentage coefficient using the 3 wheels’ average load can be seen in Figure 4. The data was very promising and was fit with a 4th order polynomial function in order to develop the needed relationship to calculate the percentage coefficients. Similar functions were also fit with 4th order polynomial functions for the maximum and minimum wheel loading cases. Once these three functions were created, all the percentage coefficients can be solved with their respective distance ratio. Finally, the wheel loading equation only contains one unknown \(W\) (largest wheel load due to the smallest distance from the CG).
Therefore, W can be solved and each wheel loading position can be found by taking its percentage coefficient times the value W.

![Graph](image)

**Figure 4:** A plot of the distance ratio versus percentage coefficient using the average wheel loading scenario.

Now that a method to determine wheel loading was developed, it was then tested to evaluate accuracy. First off, the statically determinant cases are still solved using the three static equations. Therefore, when the user selects 3 or fewer wheels, the wheel loading calculations are 100% accurate. In order to evaluate accuracy for situations with more than three wheels, two test case scenarios were run in both the ASDS software and in FEA to determine the developed method accuracy. In the four wheel test cases, an minimum accuracy of 88.37% was reached while the maximum accuracy was almost 92.4%. It was also determined that the most accurate of the three methods was using the average wheel load of the three equidistant wheels versus the maximum or minimum load. With that said, the developed method accuracy in the five wheel test case scenario was less than 50% accurate. Hence, ASDS in its current state can evaluate wheel loading with 88.37% accuracy up to four wheels. New functions are being developed in a similar manner in order to effectively evaluate loading scenarios for more than four wheels. FEA is being used to develop the functions, except different wheel position scenarios are being used in order to get more effective representations of the loading occurring in each case.

A third assessment tool for concept analysis is tipping angle. The term tipping angle refers to the minimum angle a product can be subjected too before tipping over. This is caused by the CG passing over perimeter’s line of support due to the subjected angle of, for example, a hill or road. There are several steps involved in computing the tipping angle for any product at one time.

Initially, all the supports (tires) must be specified by the user just like is done in the wheel loading scenario. Afterwards, the perimeter of these supports must be generated. Next, the shortest distance from the CG to the perimeter must be calculated. This done by first calculating the minimum point for each line of the perimeter. The perpendicular point on each line of the perimeter to the CG is calculated. If the perpendicular point lies on the line, then it becomes the minimum point for that line of the perimeter. However, if the point does not lie on the line, then whichever endpoint of the line is closer to the CG becomes the minimum distance point. After all the minimum distance points for each line in the perimeter are calculated, the smallest of these distances is then determined and is the nearest perimeter point to the CG. This can be seen quite clearly in Figure 5.

![Diagram](image)

**Figure 5:** Diagram showing the CG position and nearest perimeter point used to calculate the tipping angle.

The tipping angle is the angle between the vertical line below the CG and the line to the nearest point on the perimeter. This is the tipping angle of the product.

Now that the position of both the CG and nearest perimeter point is known, the tipping angle can be calculated. It requires using an equation to find the angle between two 3D vectors. One spanning vertically downwards from the CG, and the other spanning downwards in the direction of the nearest perimeter point.

4. **EXAMPLE**

To demonstrate the use of ASDS, a hypothetical conceptual design process is presented based on a Boeing 777. Two different concepts are going to be generated with different goals in mind. The first concept is created using legacy data while the second is generated using primitive objects and manual manipulations. The goal of concept 1 is to make a much larger passenger jet including a larger fuselage to house additional passengers and appropriately larger wings. Additional hypothetical design constraints were higher cruising speed, which would affect the swept angle of the wings and the number of engines. Figure 6 shows the results of the ASDS session for this conceptual design: the original Boeing 777 compared to the redesigned concept to be evaluated at the conceptual design stage.
4.1 ADSS Procedure

The hypothetical conceptual design exercise for concept 1 begins by loading a model of the current Boeing 777, shown in Figure 8. ADSS imports each individual part file into a pre-constructed tree hierarchy data structure represented as a scene graph on the right-hand side of the client interface. Each component is placed within a functional group created by the user. Functional groups can be manipulated at any time, and manipulation consists of group creation/deletion, filtering parts between groups, and even creating groups within groups. This particular model consists of four functional groups comprising a total of 11 individual parts.

Figure 8: The Boeing 777 original design imported into the ADSS software.

The rest of the user interface is quite simple. For every function available in the software, the user can either activate it by using the file menu structure at the top of the window, or by accessing a corresponding icon representing that function located on the tool bar.

The hypothetical conceptual design process begins by adding different wings—from an Airbus 340—into the scene graph tree structure to replace those originally built on the 777. Once the A340 wings are imported, the 777 wings are deleted from the scene graph as shown in Figure 9. Next, each wing was scaled to the appropriate size for the new design then placed into the correct position by the user using visual references. Each wing is then rotated (about the screen Z-axis) in order to produce a more aggressive sweep angle in order to meet desired goal of creating a larger plane requiring larger wings and a higher cruising speed. After the wings are complete, the user increases the overall size of the fuselage by twenty percent, allowing for more passenger capacity. This was done by using the global scaling feature on the toolbar.

Figure 7: Boeing 777 (left) and the redesigned concept 2 (right).

In a traditional conceptual design exercise, engineers would work with a variety of tools and experts to capture and convey their ideas. For example, the conceptual design team may turn to CAD, industrial design systems, image processing software, such as PhotoShop, and perhaps even graphic artists, to capture the overall intent of the concept and its configuration. This is often a tedious and time-consuming process, and the end product is generally useful only as a visual reference.

ASDS, in contrast, enables reuse of legacy product geometry, simple manipulation of part and subassembly relationships for product reconfiguration, part shape change tools, and simple geometry creation tools to facilitate quick and efficient concept generation and evaluation. In addition, all of these capabilities are enabled inside an immersive virtual environment to facilitate more accurate 3D perception. All these tools working simultaneously together helped create each of the two concept designs seen above.

Figure 6: Boeing 777 (left) and the redesigned concept 1 (right).

The goal of concept 2 is to simply evaluate wing parameters. All the features of the plane will remain the same except parameters such as sweep angle, chord length, and wing position will be evaluated in the second concept. Below in Figure 7, the original Boeing 777 can be seen from the top view next to the second conceptual design created inside ADSS.
Figure 9: Airbus 340 wings imported into the scene graph and placed in their appropriate position with different sweep angles.

The weight for the fuselage is updated by editing the meta-data tag for weight inside the CG Info tab located at the bottom right of the client UI window. This updates the CG information stored for the fuselage, and the entire product CG is recomputed in real-time as seen in Figure 10. The final step for creating the new concept design is adding two additional smaller sized engines into the scene graph, then placing them appropriately relative to the new wings. These engines were the same as those used on the Boeing 777, only scaled smaller (scaled using the global scaling tool).

Figure 10: Top view of the Boeing 777 in transparent mode (left). New concept design in a top view, also in transparent mode (right).

Once geometry creation is finished, engineers can evaluate the new design based upon the configuration of the model as well as the physical analysis feedback ADS provides. In Figure 10, above, both designs are compared against one another in transparent mode. This allows the user to view the CG location in real-time for both configurations. Clearly, the CG jumped dramatically from one design to the other. This effect can be further manipulated by editing parameters such as weight, sweep angles, wing positions, additional engines, etc.

Figure 11 shows isometric views of the original Boeing 777 and the concept redesign of concept 1 of the plane.

Figure 11: Final isometric views of concept 1 in contrast.

Figure 12 below shows the original Boeing 777 alongside the redesign for concept 2 of the plane. The redesign produced a unique concept by using different wings and sweep angles, a larger fuselage, and additional engines. ADS provides real-time feedback to the user during the entire process start-to-finish. In order to create this demonstration, approximately an hour of prep-work went into creating the initial tree hierarchy and processing the geometry files. But ultimately, what may take a design engineer weeks to accomplish can be accomplished in minutes using ADS.

Figure 12: Final isometric views of concept 2 in contrast.

5. CONCLUSIONS AND FUTURE WORK

This paper presents the architecture, implementation, and evaluation capabilities of the Advanced Systems Design Suite (ASDS) for interactive conceptual design in an immersive virtual reality environment. ADS supports the reuse of legacy data, primitive geometry creation, configuration hierarchy manipulation, and simple part transformations to facilitate simple creation of 3D product concept models. During the formation of the concept, analysis features of the software run in real-time to help the engineers evaluate the feasibility and validity of the emerging design. ADS allows for easy creation and evaluation without extensive CAD modeling and physics-based analyses. Therefore, engineers can have more time to evaluate additional concepts and ultimately, produce better results. The 3D model and assessment output for multiple concepts will foster new ideas and through multiple iterations on several conceptual designs, specific requirements and a direction for the product can then be determined.

Many additional features are currently planned for the ADS. First, the ability to import geometry created on the fly from a software package such as "Google SketchUP" will be incorporated. ADS is already capable of importing such models; however, the process has yet to be automated. This will allow for geometry creation that is not currently available from legacy data and which is more complicated than can be created using ADS primitives. FFD (Free-form deformation) (30) tools will also be implemented in order to create much more complicated geometries using simple primitive
geometries and using mouse functions to modify the shape of components. Currently, the UDP communication from the tablet to the immersive viewer is only one-way. Enhancements enabling two-way communication are currently under development.

Additional features for concept assessment are also underway. Additional physical computations will be available for properties such as tipping moment and other vehicle specific parameters that can be derived from dimensional and weight characteristics. Also, additional tools designed to allow measurements and scale to be computed and displayed will be developed. These are critical for beginning to handle issues in downstream processes such as manufacturing and packaging as well as use of a concept with other vehicles and structures (e.g., door of an aircraft lining up with a jetway). Lastly, through the use of topology optimization, the ASDS will be capable of automatically generating concepts. This will require additional user input to formulate the optimization problem. It is not anticipated that these automated results will be used directly, but rather they will point out new directions for a design team to explore to generate concepts that are not always extensions of legacy designs.

6. REFERENCES


