A system for rapid creation and assessment of conceptual large vehicle designs using immersive virtual reality

Christian Noon
Iowa State University

Ruqin Zhang
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 Technology Innovation Center

Follow this and additional works at: http://lib.dr.iastate.edu/me_pubs

Part of the Computer-Aided Engineering and Design Commons, Graphics and Human Computer Interfaces Commons, and the Manufacturing Commons

The complete bibliographic information for this item can be found at http://lib.dr.iastate.edu/me_pubs/133. For information on how to cite this item, please visit http://lib.dr.iastate.edu/howtocite.html.
A system for rapid creation and assessment of conceptual large vehicle designs using immersive virtual reality

Abstract
Currently, new product concepts are often 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 due to the complexity of these evaluation processes. Thus, promising concepts might be eliminated based solely on insufficient time and resources for assessment. 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 concepts that may have been previously abandoned due to time constraints. To address these issues, the Advanced Systems Design Suite (ASDS), was created. The ASDS incorporates a PC user interface with an immersive virtual reality (VR) environment to ease the creation and assessment of conceptual design prototypes individually or collaboratively in an immersive VR environment. Assessment tools incorporate metamodeling approximations and immersive visualization to evaluate the feasibility of each concept. In this paper, the ASDS system and interface along with specifically designed immersive VR assessment tools such as state saving and dynamic viewpoint creation are presented for conceptual large vehicle design. A test case example of redesigning an airplane is presented to explore the feasibility of the proposed system.

Keywords
Virtual Reality Applications Center, Product configuration, Conceptual design, Virtual reality

Disciplines
Computer-Aided Engineering and Design | Graphics and Human Computer Interfaces | Manufacturing

Comments
NOTICE: this is the author’s version of a work that was accepted for publication in Computers in Industry. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in Computers in Industry, [63, 5, (2012)] DOI:10.1016/j.compind.2012.02.003

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

This article is available at Iowa State University Digital Repository: http://lib.dr.iastate.edu/me_pubs/133
A system for rapid creation and assessment of conceptual large vehicle designs using immersive virtual reality

Christian Noon a,*, Ruqin Zhang b, Eliot Winer b, James Oliver b, Brian Gilmore c, Jerry Duncan d

a Department of Computer Engineering, Iowa State University, Virtual Reality Applications Center, 1620 Howe Hall, Ames, IA 50011, USA
b Department of Mechanical Engineering, Iowa State University, Virtual Reality Applications Center, 1620 Howe Hall, Ames, IA 50011, USA
c Advanced Systems Engineering, John Deere Moline Technology Innovation Center, One John Deere Place, Moline, IL 61265, USA
d Collaborative Science, John Deere Moline Technology Innovation Center, One John Deere Place, Moline, IL 61265, USA

ABSTRACT

Currently, new product concepts are often 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 due to the complexity of these evaluation processes. Thus, promising concepts might be eliminated based solely on insufficient time and resources for assessment. 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 concepts that may have been previously abandoned due to time constraints. To address these issues, the Advanced Systems Design Suite (ASDS), was created. The ASDS incorporates a PC user interface with an immersive virtual reality (VR) environment to ease the creation and assessment of conceptual design prototypes individually or collaboratively in an immersive VR environment. Assessment tools incorporate metamodeling approximations and immersive visualization to evaluate the feasibility of each concept. In this paper, the ASDS system and interface along with specifically designed immersive VR assessment tools such as state saving and dynamic viewpoint creation are presented for conceptual large vehicle design. A test case example of redesigning an airplane is presented to explore the feasibility of the proposed system.

Published by Elsevier B.V.

1. Introduction

Product design is an information intensive engineering process of decision-making. While these processes are often industry, company, or even product specific, there are activities common to most. Generally the first phase is collecting and defining design specifications about the product such as performance, quality, and safety. Next is concept generation where preliminary design ideas are proposed to meet design specifications. This is often followed by detailed design, where all design specifics such as part dimensions, material specification, and assembly arrangement, 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 aid design and analysis at the detailed stage of design [1–3], whereas concept generation and selection are still mostly dependent upon experience of engineers and use of software tools not built to handle the unique requirements of concept generation.

As mechanical systems and products increase in complexity, early stages of a design process become more critical to the success of the resulting product. Problems identified early can be fixed at much lower costs (i.e., money and time) than those found in later stages where parts, dimensions, and cost have been determined [4]. Early design stages typically include engineers identifying the requirements of a particular product and producing a concept pool using various creative methods such as brainstorming [5]. Engineers produce as many different concepts as possible to generate a wide variety of ideas to evaluate at the next level of design. Depending on the project, concept generation could produce tens to hundreds of possible concepts. Once the pool of
concepts has been established, engineers must reduce the list to a manageable number to proceed to detailed design. Conceptual design requires very easy to use tools as most individuals will not be design software experts. The tools should also foster collaboration between members since most ideas will undergo multiple iterations.

2. Background

A description of a conceptual design can be decomposed into various aspects including function, behavior, and structure [6]. To generate and select the feasible solutions, it is necessary to determine the correlations and interactions among these aspects. Computers have been used extensively in areas of simulation, modeling, and optimization, but there are relatively few applications at the conceptual design stage [7] due to the lack of knowledge of design specifications and constraints. This lack of knowledge causes two inherent difficulties in conceptual design activities: (a) modeling interactions between components and (b) reasoning to generate and select feasible solutions.

2.1. Concept selection methods

To overcome these modeling and reasoning problems, some design related techniques and methodologies have been developed. Sahin et al. [8] developed a graphical modeling tool to visualize the modeling method to address the challenges of product design decisions. Chang et al. [9] extended this work to support the graphical modeling tool with an ontology-based approach to promote the systematic capture of design knowledge. Cao et al. [10] proposed a port-based ontology to map the concept connections and interactions to compute semantic similarities. Christophe et al. [11] combined the use of function–behavior–structure, System Modeling Language, and artificial intelligence to create a dynamic mapping of ontology layers. All of these methods contain unique ways to use ontology to map the functional relationships between form and function of product designs. Understanding the internal and external functional relationships of a product can be useful for determining appropriate directions to take in the concepeting process. However, they impose a structure that might artificially constrain promising alternatives and may be more useful between the conceptual and detailed design phases. Research has also been done to try to provide more high-fidelity feedback to conceptual designers. Taskahashi et al. [12] integrated a detailed flight control systems synthesis tool into a vehicle configuration multidisciplinary design optimization (MDO) environment to better simulate aerodynamic efficiency, stability, and controllability in air vehicle configurations. Noon and Winer [13] used metamodelling techniques to capture high-fidelity analysis trends from legacy geometry datasets to provide real-time feedback of conceptual design models for large-vehicle designs. Both of these techniques provide useful ways to analyze various design configurations. However, these techniques are only applicable towards the end of the concept generation process. This is because they rely on data that is either not available or undecided in the early concept stages.

Significant research has also been performed on overall design processes such as axiomatic design [14], decision-based design [15], and specific stages of a design process such as quality function deployment (QFD) [16]. Based on Keeney’s Value Focused Thinking [17,18], Jin et al. took a value-based design (VBD) [19,20] approach to conceptual design by specifying designer’s intent with design variable values. 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. Hoyle and Chen [21] created a design tool called product attribute function deployment (PAFD), which extended the qualitative matrix principles of QFD with utilizing the quantitative decision-making processes of decision-based design (DBD). QFD is a useful tool for tracking engineering design parameters and constraints based on customer requirements and feedback. It enables design engineers to track each new concept’s ability to meet design goals and customer requirements throughout the development process. Despite these advantages, QFD does not assist in the creation of new product concepts.

Concept selection methods exist to help engineers rank a population of concepts. Examples of these methods are estimating technical difficulty, Pugh concept selection charts, and numerical concept scoring [6]. These methods have been proven effective but are simply a ranking system of engineers’ opinions on each concept’s ability to meet defined criteria of the design proposal. Indepth modeling and analysis (factual hands-on information) does not play a role in these elimination sessions. To use these methods more effectively, more information needs to be provided to the engineers to implement these methods to make concept selections and decisions.

2.2. CAD packages

Due to the complexity and information needed by detailed design tools, an adequate evaluation of every conceptual configuration cannot be performed. Such evaluations would be too time consuming and costly for most companies. To address this problem, several CAD software companies have released “lightened” versions of their products [22–24]. These products all have a common theme of trying to provide less functionality for a much smaller price. For several engineering companies, these lightened products have proved to be very beneficial. Peter Newbury, owner of Rapscallion, Inc., claimed “days of effort chasing the evolution of ideas were replaced with just one day of modeling time in Pro/E after a few hours in Pro/CONCEPT clarified the concept with both the designer and engineer.” Pro/CONCEPT [25] is a software product developed by PTC specifically for modeling extreme product ideas without constraints. There are several other success stories of these stripped down applications providing companies with tools more suited to the job of conceptual design. However, all of these application’s interfaces are still very complex and offer many options and features. They are useful for bridging the gap between multidisciplinary design teams, but generally these tools ship with large tutorial sets, help documents, and customer service recommendations. A non-CAD user generally is still unable to use these applications. Therefore, the majority of design teams still require a CAD expert to run the software. One additional downside to these de-featured software packages is that all of the analysis tools have been stripped out, so there is no way to validate a design.

2.3. Virtual reality

Nowadays, VR technology is gaining increasing utility for a variety of applications in product development or virtual reality aided design (VRAD) [26]. Companies all over the world are now using VR for ergonomic studies, virtual assembly, and factory floor planning. VRAD has also entered the design processes of automotive and aerospace industries. Some of these applications have been described as being very useful when compared to conventional CAD methods and in many cases there are advantages to integrating VRAD into the design process. In summary VRAD adds the potential for time-to-market reductions, quality enhancements, and cost savings to the product development cycle [27]. Zimmermann also elaborates on the difficulties in integrating CAD and VR display systems in terms model preparation. Raposo et al. presented the ENVIRON (Environment for VIRTual Objects Navigation) system to
act as a CAD to VR data exchange tool by decreasing file size, enhancing material properties, and re-tessellating complex NURBs surfaces of the CAD file to produce an enhanced VR representation of the data [28]. Later Kim and Weissmann presented the MEMPHIS (Middleware for Exchanging Machinery and Product Data in Highly Immersive Systems) system, a data exchange middleware integrating PDM, CAD, and VR systems through common interfaces. The MEMPHIS system accomplished this data exchange by using PDM adapters to serve as common interfaces between the systems allowing VR-specific metadata to be maintained between systems as well as common CAD to VR format file conversions [29]. Both the ENVIRON and MEMPHIS systems present useful solutions to the common data exchange issues between CAD and VR environments and file formats.

With real-time interactive graphics, stereoscopic displays, and user tracking, VRAD has also been utilized in applications where one-to-one scale is important or when the assessment of complex geometric relationships is required. Haptic interfaces have also been employed with VRAD for assisting conceptual design [30]. Fischer and Vance [31] used haptic devices inside a six-sided virtual reality environment for installing an aircraft rudder pedal assembly. Duncan and Vance [32] later developed an immersive virtual reality environment to help engineers better understand the parametric data stored in the model system. The authors also created a collaborative virtual environment (VRoom) that allowed designers from multidisciplinary backgrounds to view and manipulate 3D models in an immersive environment simultaneously. With all these technologies available, immersive VRAD can provide a collaborative design environment with additional features, which cannot be matched with a 2D desktop environment.

2.4. Opportunity

Concept selection methods and CAD packages have their advantages and disadvantages. All methods have numerous capabilities but, in today’s digital age, still do not define a clear set of tools to be easily integrated into an existing design process. In practice, conceptual design is still done with spreadsheets and sketches, and is collaborative in nature. Design teams typically consist of cross-functional stakeholders that represent a wide variety of backgrounds in order to bring unique perspectives to the concept generation process. The goal of this research is not to reinvent the conceptual design process but to assist in its current form.

The lack of available tools and the diversity of conceptual design teams helped define the research goals of this work. The first goal was to create an environment to support rapid idea creation as well as actual 3D model generation. Another goal was to provide a 2D CAD-like environment as well as an immersive VR environment for conceptual design teams to collaborate within. Each environment would provide its own unique way of interacting with the same 3D model. These different perspectives could help foster additional ideas. The final goal was to provide designers with a set of evaluation metrics to determine whether concept was feasible. It would also be useful if these evaluation metrics could update in real-time so designers were aware of the effects each decision had on the overall performance of the concept. Current CAD design analysis takes an alternative approach requiring the design to be finalized before running a high fidelity analysis. These research goals could combine to form a means to quickly create and analyze lower fidelity digital models in real-time to make accurate and informed decisions as early as possible in the design process.

This paper presents a conceptual design system, which reuses legacy data for geometry creation and real-time assessment. This is enabled by collaborative desktop and immersive VR environments. The following section of the paper presents the methodology of constructing the ASDS system. This section focuses on the architecture, interface design, assessment tools, and immersive capabilities. Then, a section detailing specific immersive assessment tools is presented. Finally, a test case is presented followed by conclusions and future work.

3. ASDS methodology

The software is named the Advanced System Design Suite (ASDS) [34,35]. The ASDS was created to enable designers to work collaboratively to quickly build a 3D model of a proposed design, assess a concept in real-time using metamodel approximations, and visualize the results on both desktop and immersive VR systems. The user interface was designed specifically for allowing non-CAD experts to create a design by removing some of the time-consuming requirements, which CAD requires for maintaining boundary representations. ASDS allows designers to reuse both geometric and analytic legacy data to enhance concept generation productivity. The geometric models (usually CAD models) are typically de-featured to a level of detail appropriate for conceptual design. Once the models have been de-featured, they are converted into a format readable by the ASDS system if necessary.

3.1. Architecture and operation

The underlying architecture of the ASDS system is divided between the desktop and immersive applications. The desktop application runs on multiple operating systems including Windows, Mac OS X, and Linux on both 32 and 64-bit platforms while the immersive application operates only on 64-bit Linux operating systems. The system utilizes several open-source packages including OpenSceneGraph [36] for scene graph management and rendering, wxWidgets [37] for developing the user interface of the desktop application, and VR Juggler [38] for VR display clustering and interaction device abstraction.

To facilitate an immersive design session using the ASDS system, a designer drives the design session through the desktop application while a second user controls the immersive application using a gamepad controller. All the design actions are controlled in the desktop application which include anything from loading legacy CAD models, loading primitives, reorganizing the model hierarchy, manipulating part orientation, changing material properties, performing measurements, and using assessment tools. Once a design action is performed on the desktop, the change is transmitted over a network connection to the immersive application where the modification is replicated in real-time. Both applications contain the same geometric design at all times, however each application can independently control its own view of the design. This is handled by standard view controls in the desktop application and navigating with a gamepad in the immersive application.

All the legacy data (e.g., part libraries) acquired from previous designs is stored in a location accessible by both applications. In its envisioned use, the ASDS part libraries will consist of legacy product models comprised of geometric as well as physical properties (metadata) of previously created products. The parts library may also contain new parts created by other programs such as Google SketchUp, which are not part of the corporate legacy database as well as one of many standard primitives that are pre-installed with the ASDS system. The physical data embedded within the original CAD files can be transferred into the ASDS system at any time. Additionally, if the metadata did not originally exist in the CAD files, each property can be entered into the ASDS system directly by engineers with knowledge of the models. The metadata properties can then be used for further evaluation and analysis. Metadata currently consists of part weight and center of
Gravity (CG) information, which is used to enable all assessment calculations.

3.2. Interface design and interaction

When designing the user interface and interaction models for the ADS system application, the focus was to free designers from complex geometrical constraints to ease geometry creation for all levels of users. In order to accomplish this, a different design philosophy was adopted. Instead of implementing the common CAD theme of design by starting with a blank slate and constructing each part and feature through complex sketches, extrudes, revolves, and other common CAD geometry creation techniques, a much simpler approach was taken. By providing legacy geometry as well as an extensive library of primitive shapes, designers can conceptually construct geometric models of their idea through various combinations of the parts available. Even though users cannot create as detailed models as they could with the same CAD system, they are able to create a rough representation of their idea suitable for this stage of design.

Another way the ADS system allows users to quickly create concepts is by eliminating the need for complex mating schemes and precise dimensioning which CAD systems require. Instead, the ADS system is constructed in a manner where parts can be translated, rotated, and scaled in any direction or orientation without regards to other objects in the design. An example of the scale manipulation tool can be seen in Fig. 1. The parts in the assembly are only identified by their location in the scene graph instead of also requiring precise relative location and association information with adjacent parts. The application was designed to handle this type of drag-and-drop philosophy of design for all aspects of hierarchy management, material properties, metadata handling, and assessment tool calculations. These interface and interaction design decisions allow the ADS system to allow fast concept generation with a small learning curve.

To create a concept using the ADS system, the first step is to import geometry into the scene graph. This is accomplished by loading data from various sources including CAD geometry from legacy designs, new geometry, or geometric primitives. It is also important to note that these parts do not need to be extremely complex at this stage of design. All ideas are merely proof-of-concept until engineers dwell down through multiple iterations to settle on a specific design. Once geometry has been imported, it needs to be organized into a sensible hierarchical structure. The interface has a standard CAD assembly tree widget for managing each functional group of the assembly. Designers can add new groups as necessary and reorganize parts and groups through drag-and-drop functionality. This can all be accomplished quickly and is very intuitive as it matches most interface models of today's software.

The most important feature of the user interface is the ability to quickly manipulate the parts and groups within the concept. The manipulations are made possible through the use of 3D manipulation tools for translation, rotation, and scale. A group or part can be selected and manipulated anywhere in 3D space using any of these three methods. Again this design philosophy enables users to avoid the complexities of today's CAD system assembly tools. Once the concept is completed, a designer can save the design out in the ADS system's native file format which wraps an XML description of the scene graph on top of the loaded geometric models. By doing so, the ADS system is completely non-destructive as well as achieves very efficient write times when saving concepts. By developing each feature of the interface to facilitate fast geometry creation for all users, the ADS system provides designers with an effective means for constructing many geometrical concepts.

3.3. Assessment tools

A 3D visual representation of a new concept design is, in itself, extremely useful, but additional assessment tools are needed to
properly judge a concept’s feasibility. The goal here was to provide engineers with useful and concise information in real-time. With this goal in mind, mathematical assessment tools have been integrated into the ASDS system.

3.3.1. Center of gravity
The first assessment tool dynamically computes the CG of the entire model in both the desktop and immersive applications as seen in Fig. 2. Individual weight and CG positions are stored as metadata within the scene graph. With this information, all the part transforms can be concatenated to compute the CG location for any individual part, subassembly, or entire concept design. Eqs. (1) and (2) allow \( W_n \) and \( P_n \) (weight and position) to be solved directly:

\[
W_n = \sum_{i=0}^{n} W_i \quad (1)
\]

\[
P_n = \frac{(P_{n-1} \cdot W_{n-1}) + (P_{n-2} \cdot W_{n-2})}{W_{n-1} + W_{n-2}} \quad (2)
\]

To visualize the CG location, its position is represented by a red sphere inside the scene graph while the selected model turns transparent. The transparent view is required to view a CG position hidden inside part of the scene graph. The sphere position updates in real-time when changes are made in the desktop application.

3.4. Tipping angle

A second assessment tool computes the static tipping angle of the entire model. The term tipping angle refers to the minimum angle a vehicle can be subjected to relative to a given ground plane before tipping over. To calculate the tipping angle, support points – wheels, legs, etc. – keep the model in contact with the ground must be selected. By clicking the tipping angle button, the ASDS system uses the overall CG and contact positions of the support points to calculate the minimum tipping angle of the model, see Fig. 3. To calculate the tipping angle, first the ASDS gathers the positions of each selected wheel in addition to the CG position. Then, the perimeter lines of the support points are calculated using search algorithms. The nearest perimeter point is then determined using both Eqs. (3) and (4). Eq. (3) calculates the perpendicular distance between point \((x, y, z)\) to the line \(Ax + By + Cz + D = 0\). Eq. (4) calculates the distance between the CG position and each end point of each perimeter line.

\[
d = \frac{|Ax + By + Cz + D|}{\sqrt{A^2 + B^2}} \quad (3)
\]

\[
d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (4)
\]

Then the point with the minimum distance from the CG position is calculated. Finally, the tipping angle is calculated as the angle between the CG vertical line and the line to the nearest point on the perimeter using Eq. (5).

\[
\tan(\theta) = \frac{d}{d_c} \quad (5)
\]

The information is presented graphically by showing the tipping angle, perimeter lines, and tipping angle position. The tipping angle value is also displayed in an assessment panel. If the design at any point is completely tipping, the assessment panel alerts the user with intuitive error messages describing why the vehicle is now tipping in its current configuration.

4. Wheel loading

Another assessment tool built into the ASDS system is wheel loading. The term wheel loading comes from our target application to ground vehicles, but “wheel” simply refers to any support point. This tool first requires each support point be selected just like the tipping angle tool. Once completed, the ASDS desktop application uses the contact positions of the support points and the overall CG position and weight to calculate the load distribution for each selected object. This information is then displayed graphically in both the desktop and immersive environments. This calculation is accomplished in one of two ways. If three or less support points were chosen, the statically determinant problem can be solved by summing the forces in the \(y\)-axis and moments in the \(xz\)-axes. Eqs. (6)–(8) allow \(F_x\), \(F_y\), and \(F_z\) to be solved directly:

\[
\sum F_x = 0 = F_1 + F_2 + F_3 - CG \quad (6)
\]

\[
\sum M_x = 0 = \pm F_1 a_x \pm F_2 b_x \pm F_3 c_x \pm CG d_x \quad (7)
\]

\[
\sum M_z = 0 = \pm F_1 a_z \pm F_2 b_z \pm F_3 c_z \pm CG d_z \quad (8)
\]

This is very useful, but many of the ground vehicle concepts the ASDS is being designed for consist of more than three wheels creating a statically indeterminate problem. The second method implemented to solve this problem was to build polynomial response surface (PRS) metamodels [39] from simulated FEA simulations performed on a simulation model—the loading rig seen in Fig. 4. The loading rig was developed in ABAQUS [1] and consisted of several pieces including the base, legs, and supports. The base was located in the center of the loading rig and attached...
to each of the four legs. The CG force was also applied to the top face of the base. The legs of the loading rig exist only as supports to distribute the CG force to the reaction forces of the supports. The legs were four meters long, which was just long enough to support the variable support positions, which varied according to the parameters of the random sample array. The supports reside on the bottom of the legs and were fixed so FEA could calculate the reaction forces at each of the four support locations. The loading rig had steel material properties to keep the structure rigid, and with such a low CG force, deformation was assumed to be negligible. Even though this is a simple model, the loading rig is appropriate for measuring performances relative to conceptual design. It should be remembered that precise measurement here is not a criterion for this stage of design. Rough estimates of the vehicle properties for all concepts are more beneficial than exact measurements of a single concept.

To generate the wheel loading data for a generic large vehicle, four wheels with variable positions and a static center position made up the design variables. Each reaction force was then calculated as a percentage of the CG force. Since the entire design space consisted of thousands of FEA simulation design points, a random sampling of points throughout the design space was used to generate 50 design points, a number experimentally determined to be adequate to fully represent the design space. An additional 30 points were generated for evaluation of the metamodel accuracy.

After the datasets were generated, second and third-order PRS metamodels were constructed. A generic second-order PRS metamodel can be seen below in Eq. (9):

\[
W = c_1 + c_2 x_1 + c_3 x_2 + c_4 x_3 + c_5 x_4 + c_6 x_1^2 + c_7 x_1 x_2 + c_8 x_2 x_3 + c_9 x_3 x_4 + c_{10} x_1^3 + c_{11} x_1^2 x_2 + c_{12} x_1 x_2^2 + c_{13} x_2^3 + c_{14} x_2 x_3 + c_{15} x_3^2 + c_{16} x_3 x_4 + c_{17} x_4^2
\]

Least squares regression was used to solve for the coefficients. After the PRS metamodel was generated, the coefficients were calculated and then a polynomial expression remains to quickly calculate the output variables. An example of a second-order PRS metamodel calculated can be seen below in Eq. (10):

\[
W = 0.12 - 1.03 x_1 + 0.72 x_2 + 0.11 x_3 + 0.59 x_4 + 0.76 x_1^2 - 0.24 x_1 x_2 - 0.19 x_1 x_3 - 0.10 x_1 x_4 - 0.28 x_2^2 + 0.22 x_2 x_3 - 0.30 x_2 x_4 - 0.35 x_3^2 + 0.51 x_3 x_4 - 0.36 x_4^2
\]

The accuracy of each model created was computed against the second test dataset, independent of the one used to build the model. When testing the results of the metamodels against the actual FEA analysis data, the root mean square error (RMSE) values were measured with no more than seven percent error. This meant the metamodels were producing results that were approximately 93% accurate. The loading rig design has been shown to several large vehicle industrial designers who all agree it is a good way to assess these conceptual design parameters. These industrial designers also agree that 93% is plenty accurate to provide sufficient data to help make informed decisions at the conceptual stage of design.

These mathematical assessment tools are very helpful and serve their purpose in both the desktop and immersive ASDS applications. However, these assessment tools alone cannot take advantage of specific visual capabilities of the desktop and immersive VR applications. These tools take advantage of the visual capabilities in the ASDS system to allow design issues to be determined early and potentially save time and money further down the design process.

4.1. Virtual measurement

A popular CAD feature known as mating was eliminated in the ASDS system. Mating requires the user to fuse different parts together in order to maintain correct dimensionality with respect to the entire product. The ASDS has completely eliminated this feature to increase productivity by not requiring exact dimensioning at every step of the creation process. Instead, a visual measuring tool was integrated into the ASDS desktop and immersive applications for occasions when designs must meet...
specific requirements (e.g., must be 36 in. off the ground) at the conceptual design stage. When design teams collaborate within the immersive application, quick dimensional information is key to making rapid drag-and-drop part manipulations to verify whether different part library objects can be swapped to create a new dimensionally sound concept. The virtual ruler system allows the user to manipulate components and return physical characteristics from the scenegraph. Once the user selects a geometric boundary, both applications return physical characteristics of the selected boundary such as radius, length, width, etc.

4.2. Snapshots

A project management tool referred to as “snapshots” was developed to save the current configuration within the same ASDS project file (.xcd file). All that is required to create a snapshot is to provide a name, and the ASDS system saves the entire configuration for later use. This can be extremely useful for documenting the concept design process in addition to capturing specific base configurations for which to build upon in the future. Each snapshot generates a preview thumbnail so the user can cycle through various snapshots visually instead of just referring to the title.

In order for the ASDS to save various configurations, the scenegraph hierarchy and part transforms are copied into a temporary text file where all the current scenegraph information is stored. Afterwards, if a previous snapshot needs to be recalled, the user clicks on a previously generated snapshot, which can be seen on the right of Fig. 5. The ASDS then finds the scenegraph information stored for the selected snapshot, and updates the current scenegraph to match the information saved inside the temporary text file and design can continue. Snapshots are also stored within ASDS project files. Upon reopening an .xcd file, all the snapshots are reloaded.

The ASDS already allows users to make hundreds of changes quickly in the environment. This tool allows the design team to capture the important configurations along the way to develop multiple concept iterations and branches from a particular saving point. Design teams can save different configurations during a design session, whether it be to save a viable concept configuration, a stable configuration to which multiple branches will be created from later, or a final concept to possibly move forward with into the next stage of design. Additionally, snapshots allow engineers to look back throughout the design stages to track the development process of the concept. This can be useful to others to see how a concept was redesigned from the original.

4.3. Dynamic viewpoints

A feature built strictly for the immersive design session is the ability to dynamically set a viewpoint anywhere throughout the scenegraph to the user head position. First, the user navigates the immersive scene with a typical two stalk game controller. After the user navigates to the exact location of the viewpoint, a simple button click caches the viewpoint and can be recalled at any time. The user can also cycle through the set viewpoints. Once the user toggles to another viewpoint, the virtual scene animates between the current viewpoint and the toggled viewpoint. These viewpoints can then be written out for future design sessions if so desired.

An example showing the need for dynamic viewpoints is evaluating anthropometric data. Anthropometric data requires exact positioning of specific body locations. From there, design teams can effectively evaluate design constraints ranging from seat height and steering wheel position to door handle and maintenance accessibility. This feature coupled with head-tracking technology creates a very powerful evaluation tool for an immersive VR environment. Design teams can effectively evaluate

![Fig. 5. A screen capture of a helicopter concept in the ASDS with six different snapshots.](image-url)
multitudes of positional constraints from a consumer perspective of a virtual model.

4.4. Integration challenges

One of the main reasons this system is so unique is because of all the different technologies and research that were integrated to create a single, unified conceptual design system. One of the first integration challenges was combining the use of a graphics desktop application to control an immersive application. This required coupling both wxWidgets and OpenSceneGraph to a VR Juggler and OpenSceneGraph immersive application through the use of networking. Since there is no system in any of these open-source APIs for accomplishing this type of control, a complete networking system was designed to allow control of the immersive application using the desktop. Additionally, since sending the scenegraph over the network with every change is not practical due to bandwidth limitations, the networking scheme had to be designed to be very lightweight in order to accomplish real-time updates. Each network transmission was highly optimized to send the minimal required amount of information necessary to reconstruct the modification in the immersive application.

A second major integration challenge, which arose, was the issue of navigating the immersive application. Initially, the desktop user controlled the view of the desktop application as well as the immersive application. It became apparent that requiring the desktop user to control the immersive application view was hindering their ability to focus on the main task at hand, designing the concept. For example, a viewpoint, which allowed users in a multi-wall immersive environment to “walk” around the design, resulted in the desktop user being inside the concept. This mismatch between different screen sizes and multiple screens from immersive to desktop caused many problems in creating and assessing designs. Therefore, the system had to be redesigned to allow independent views in both applications. The immersive application was modified to allow an immersive user to control the view through the use of a gamepad. Several interaction techniques were investigated with the gamepad being most useful in an ad hoc manner. Decoupling the immersive viewpoint from the desktop application produced two large improvements to the software suite. First, the desktop user was able to focus on designing the concept rather than concern themselves with the immersive application’s view. Second, the user experience in the immersive application was greatly improved due to their new ability to explore the design as they see fit without having to depend and possibly wait on others to modify their view.

Another major integration issue was the ability to update all the assessment tools in real-time based on geometric manipulations. Again, none of the open-source libraries used have support for this type of functionality. Therefore, both a monitoring system and an assessment update pipeline were devised to accomplish the task. The monitoring system is continuously watching for modifications to the design, which cause the state of the assessment tools to be out-of-date. Once the monitoring system observes these changes, each specific change is passed off to the assessment pipeline for processing. Initially, the CG calculation is updated, which then starts both wheel loading and tipping angle updates. Finally, once all calculations have been updated, the assessment tools notify the user interface to update the displayed values in the assessment panel.

Merging all of these technologies and various areas of research presented many issues, as most of these technologies are not designed to work well with others. Each integration challenge presented the opportunity to construct novel and unique solutions for the ASDS system. Due to these cumbersome challenges, integrating all these technologies and methods into a single unified system was a great success.

5. Design and assessment example

To demonstrate, the use of the ASDS, a hypothetical conceptual design process is presented based on a Boeing 777. The design criteria for the new concept is to make a much larger passenger jet including a larger fuselage and wings to transport additional passengers. Additional design constraints require a higher cruising speed, affecting the wing sweep angle and the number and size of the jet engines. The results of an ASDS conceptual design session for this idea are shown in Fig. 6.

The hypothetical Boeing 777 conceptual design exercise begins one of two ways. First, an engineer can launch the application on a desktop computer. A second way to begin the exercise is to gather the design team into a VR system with both the desktop and immersive applications running for a collaborative conceptual design session. Either of these methods can be used to create concepts individually or collaboratively with a design team. Once the desktop application is loaded, the original Boeing 777 model is imported into the ASDS application(s). Individual part files are loaded into functional groups, which are created in minutes and displayed on the right-hand side of the interface as seen in Fig. 7. Each component lies within a particular functional group and can be rearranged at any time. This particular model contains 13 individual parts contained within four different functional groups.

The design process starts by adding cube primitives to replace the original Boeing 777 wings. Once the new wings are moved into the WINGS functional group, the original Boeing 777 wings are deleted from the scenegraph. Next, each wing is scaled to the appropriate size to meet the specific design criteria. They are then placed into the correct position using visual references instead of time-consuming

Fig. 6. A Boeing 777 (left). An ASDS redesigned concept for the Boeing 777 (right).
mating and dimensioning schemes. Next, each wing is rotated about the screen z-axis to generate a more aggressive sweep angle and allow higher cruising speeds. Once the wings are complete, the user scales the fuselage by 20%, allowing for more capacity.

The fuselage weight is updated by editing the metadata tag for weight inside the Properties Panel on the left middle portion of the screen as seen in Fig. 7. This updates the CG information stored for the fuselage, and the entire scenegraph CG is recomputed in real-time. The final step for creating the new concept design is adding two additional smaller sized engines into the scenegraph, then placing them appropriately relative to the new wings. To create these smaller engines, two additional Boeing 777 engines are imported into the ENGINES functional group and scaled down to the correct size. Once the engines are positioned, the physical configuration of the new concept is finished and the assessment tools can be used to evaluate the concept feasibility. If however, different types of engines were going to be used, primitives could be substituted as placeholders until geometry for the engines was built or acquired. These primitives could then be given the same metadata parameters such as CG weight and position to continue evaluating the design without the actual engine geometry.

Once geometry creation is finished, engineers can evaluate the new design based upon the configuration of the model as well as the assessment tools provided. Enabling the CG location allows the user to visualize how editing parameters such as weight, sweep angles, wing positions, or adding additional engines affects this property. In addition to CG calculation, the other assessment tools can also be used in combination to determine concept feasibility. To calculate tipping angle as seen in Fig. 8, the user selects the wheels and then clicks on the tipping angle computation checkbox. The ASDS then returns the smallest angle at which the design will tip in a static position. If the tipping angle is too small, the CG position is too close to the landing gear and the design is infeasible.

If the tipping angle meets design specifications, the design team can then move onto evaluating the exact position of the CG in relation to the wheel loading at each of the landing gear. To calculate the wheel loads, the user selects each of the wheels of the concept design, and then the ASDS returns the wheel loading for each of the support points through either a static force and moment calculation or metamodel estimation as seen in Fig. 9.

Wheel load then allows designers to verify whether the design maintains feasibility from a CG and wheel loading standpoint. If the wheel loading is too large for a single set of landing gear to handle the weight distribution, either the design is infeasible or needs to undergo another design iteration to decrease wheel loading in high stress areas.

This simple demonstration of the capabilities of the ASDS system reinforces how well the software suite meets the research goals associated with this project. The system interface was
constructed to allow designers to make major modifications to geometry to produce entirely new design concepts with ease. ASDS also provides design teams with real-time assessment information about the design metrics of the concept based on areas of center of gravity, wheel loading, and tipping angle. These tools allow design teams to immediately see how their decisions affect the overall performance of the concept to catch problematic areas of design sooner. Finally, the system also provides 2D and 3D immersive VR environments for design teams to collaborate and use different visual perspectives to foster new ideas. One last advantage which was not demonstrated particularly well in this example is the ability of the immersive application to provide assessments not possible in 2D environments. These include the ability to sit inside the cab of a vehicle to evaluate visibility constraints, control accessibility, and even user interface mockups of touch screens. By allowing various members of design teams to use the immersive environment in these various manners, the design team is able to collaborate in a way not possible without the use of immersive technology. The design team experiences the 3D model by actually standing next to, within, or even interacting with controls. This system provides a much more interactive experience for users, especially those in the design team with less engineering experience such as management or marketing.

6. ASDS and Solidworks comparison

The previous example demonstrated how the ASDS system could be used to heavily modify an existing design as well as use the assessment tools to determine whether or not the new design was feasible. This does not, however, reinforce all the tradeoffs between using the ASDS system in place of commercial solutions in existence today. To demonstrate these tradeoffs, a double bearing assembly as shown in Fig. 10 was created in both ASDS and Solidworks. The ASDS assembly was constructed solely of primitives, which were imported from the native ASDS primitive library. An important note is that this assembly could be created many different ways with both programs. This example demonstrates one particular way to generate this assembly using both programs for comparison purposes.

The first step to creating the assembly in both programs was to create the Front Base, which can be seen below in Fig. 11. Creating this part in ASDS involved several steps. The first step was to import a cube primitive and manipulate it to the appropriate size. Next, a half cylinder primitive was imported, scaled, and positioned appropriately. To create the second cylinder, the first cylinder was copied and pasted, then positioned. This can be seen below in Fig. 12. To square up the outside of the Front Base with the end of the cylinders, another cube primitive was imported and positioned on one end of the Front Base to line up with the outside of the large cube and both edges of the cylinder. This second cube was then copied and placed on the adjacent side of the Front Base. The final step was to fill in the eight gaps between the cylinders and the cubes. To accomplish this, a trapezoidal prism primitive was imported and scaled appropriately to fit in the gap. The prism was then copied seven other times and repositioned to fill the other gaps in the Front Base.

Constructing the same assembly in Solidworks was a very different process involving three different combinations of sketches, extrusions, and extruded cuts. This process was very different from that of ASDS. This is due to the fact that the user was required to think as a CAD designer using dimensions and various combinations of sketching techniques to create the necessary sketches instead of relying on primitive shapes and visual cue manipulations. It was much easier to create the base object using a single sketch extrusion operation in Solidworks than having to fill the gaps with other primitives in ASDS. However, it was more challenging to create the precise extruded cut sketches in Solidworks than it was to resize the separate component cylinders in ASDS.

The next step of the process was to create the narrower Back Base of the assembly. To do this in ASDS, first the Front Base was copied, pasted, rotated 180°, and repositioned. Making the object narrower required scaling the three cube primitives and then repositioning them. Creating the Back Base in Solidworks was a similar operation. First, the Front Base part was saved out as the Back Base. Next, the Back Base original sketch was modified to be narrower. This then updated the rest of the part extrusions to match. Thus, creating the Back Base was slightly easier in Solidworks since only the original sketch needed to be modified instead of three different cube primitives.

Please cite this article in press as: C. Noon, et al., A system for rapid creation and assessment of conceptual large vehicle designs using immersive virtual reality, Comput. Industry (2012), doi:10.1016/j.compind.2012.02.003
Three other parts required to construct the assembly seen in Fig. 13 were the Angular Bracket, the Main Bracket, and the Bolt. In ASDS, constructing the Angular Bracket consisted of merging a trapezoidal prism and cube primitives with appropriate scaling. The Main Bracket was a cube primitive scaled to the correct dimensioning. The Bolt was a single cylinder primitive scaled appropriately for head of the bolt which was then copied, scaled, and repositioned to form the shaft.

Building each of the three parts in Solidworks was somewhat similar. Creating the Angular and Main Brackets was straightforward, as both objects required only a single sketch and extrusion. The Bolt required two sketches and extrusions. To compare the creation process of these three objects between the two programs, they were both similar. The Angular Bracket was slightly easier to create in Solidworks, but the other two were slightly faster using ASDS. A noticeable difference, however, became apparent when using Solidworks. Solidworks requires the user to know the dimensions of the part before constructing it. If the user does not know the dimensions, then they must refer back to the other parts created and use the dimensioning tools to figure out what the dimensions of the new part need to be. The ASDS system does not require the user to enter dimensions for any object. If the user wants to model to be roughly to scale, then the initial object brought into the scene can be sized appropriately using the measurement tools. Afterwards, the user can size new objects in reference to the initial object instead of having to know the exact dimensions of every object.

Fig. 12. Screenshot of the Front Base of the Double Bearing Assembly being constructed.

Fig. 13. Screenshot of the Double Bearing Assembly in ASDS with labels for each of the parts.

Please cite this article in press as: C. Noon, et al., A system for rapid creation and assessment of conceptual large vehicle designs using immersive virtual reality, Comput. Industry (2012), doi:10.1016/j.compind.2012.02.003
The final step was to generate the assembly. ASDS allows users to quickly group a set of components and reposition them anywhere in the scene. Therefore, constructing the assembly from the parts was fast and easy with the use of copy/paste and the lack of mating constraint requirements. Solidworks, however, requires each individual part to be imported into the scene, then positioned and mated to the existing assembly. There are several downsides to this approach. The first downside is that all of the mating constraints must be precise. If the parts do not line up correctly, small pieces get in the way, or the dimensions do not quite match, then the user cannot mate the two objects. The original part would need to be modified, then the assembly regenerated. Unfortunately, this process can cause previous mating constraints to be invalidated, thus requiring additional work to restore the state of the assembly with the modified part. Another downside to this approach is that once the assembly is built, it is very difficult to start taking it apart and swapping in different components in their place. This is due to the process of using mating constraints to build the assembly.

To demonstrate this challenge, the Front Base was separated from the Back Base in both applications, which can be seen in Fig. 14. Accomplishing this in ASDS required the user to create a Back Base group, use drag and drop to move all the Back Base components into the group, then translate the group away from the assembly. ASDS can then quickly swap out the Back Base set of components with a different set for design team evaluation of multiple concepts. However, when attempting this same operation in Solidworks, the limitations of fast assembly modification became apparent. In order to attempt this, the mating constraint between the Front and Back Base needed to be removed. However, when attempting to do so, many other mating constraints the assembly contained were invalidated. Therefore, after removing the constraint, other constraints were removed as well. Then when using the move tool on the Back Base, not all of the objects moved with the Back Base because the mating constraints keeping the assembly together were lost. This is just a quick example of how using visual cues, drag and drop restructuring, and basic manipulators is much more suited for rapid assembly configuration and modification than typical CAD package mating constraints.

To summarize this comparison, Solidworks proved to be better suited for creating more complex pieces of geometry such as the Front Base. However, parts, which could be represented solely by a manipulated primitive, were much easier to create using ASDS. This is not surprising, as ASDS was not designed to be a CAD replacement, but a conceptual design CAD alternative. The main advantage of using ASDS was apparent when assembling many parts into a single assembly. The user was not burdened by dimensions and mating constraints, but instead was able to use visual cues to quickly place objects in roughly the correct locations. Solidworks requires users to create each part with precise dimensions, which in turn requires users to spend more time switching contexts between parts models. Additionally, ASDS allows a user to quickly disassemble and rebuild an assembly with different parts. Solidworks mating constraints were not designed to support such quick reconfigurations.

7. Conclusions and future work

This paper presents the Advanced Systems Design Suite (ASDS), which combines many different technologies, complex mathematical techniques, and user interaction design philosophies to provide a new and unique system for facilitating 3D concept generation and assessment for large vehicle conceptual design. The use of both desktop and immersive environments with tracking and audio feedback allows users to push design decision boundaries. These technologies present the ability for designers to solve engineering problems in more effective and creative ways, as opposed to traditional methods. The immersive VR environment also allows design team members to sit inside of a cab of a vehicle to evaluate visibility constraints, control accessibility, and even user interface mockups. Mathematical approximations, or metamodels, have been incorporated to offer feedback to designers as to whether a design is feasible. These metrics operate in real-time allowing users to monitor how each design decision affects the overall performance metrics as well as identify critical design issues. Finally, the user interface was created to free designers from complex geometrical constraints such as mating and precise dimensioning as these are not necessary at the conceptual design phase. Instead, users place objects visually with respect to others without regard to small dimensional accuracies. In the event higher precision is required, the ASDS system offers a set of measurement tools to accommodate designers. The combination of all these features into a single system tailored specifically for large vehicle conceptual design offers a new and unique approach to the conceptual design process.

Many additional features are currently planned for the ASDS. First, free-form deformation [40] tools will be implemented to allow more complex geometries using simple primitives and lattice manipulators to morph the components into more complex shapes. Another tool to be integrated into the ASDS is the ability to determine blind spots from the perspective of a vehicle operator or passenger eye position. Other conceptual design assessment tools are also underway. More precise measurement tools are being designed to allow for higher accuracy dimensioning. These are critical for handling 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 jet way). Finally, through the use of topology optimization, the ASDS will be capable of automatically generating seed 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 be seeds of new directions for a design team to explore to generate concepts that are not always extensions of legacy designs.

References

Christian Noon is currently a Ph.D. candidate majoring in computer engineering and human computer interaction at Iowa State University in the Virtual Reality Applications Center (VRAC). He received his B.S. degree in mechanical engineering and human computer interaction from Iowa State University in 2008, and his B.S. degree in mechanical engineering from Iowa State University in 2006. His research interests include various software and human computer interaction technologies including computer graphics, virtual reality, software engineering, computer-aided design, collaborative conceptual design methods, and modeling and optimization techniques.

Ruqin Zhang received a B.S. degree in mechanical engineering from Beijing University of Aeronautics and Astronautics (China) in 2002, and a M.S. degree in mechanical engineering from Iowa State University in 2006. Now he is a last-year Ph.D. student of human computer interaction at the Mechanical Engineering Department, Iowa State University. His research interests include computer graphics, geometric modeling & processing, 3D visualization, nano-scale manufacturing and molecular simulations.

Eliot Winer is currently an associate director of the Virtual Reality Applications Center and an associate professor in the Department of Mechanical Engineering at Iowa State University. He received his Ph.D. and M.S. degrees in mechanical engineering from the University at Buffalo in 1992 and 1988, and his B.S. degree in aerospace engineering from Ohio State University in 1992. He teaches courses on mechanical systems design, optimization, and professional ethics. His research interests include Internet technology for large-scale collaborative design; visualization and interaction with medical data studies; multidisciplinary design synthesis; computer-aided design and graphics; and virtual reality modeling for conceptual design.

James Oliver is a Larry and Pam Pithan professor of mechanical engineering at Iowa State University (ISU). He serves as director of the Virtual Reality Applications Center, and ISU’s Interdepartmental Graduate Program in Human Computer Interaction. His research activities encompass a wide array of human computer interaction technologies, including computer graphics, geometric modeling, virtual reality, and collaborative networks for product development and complex systems. Previously, he held several executive positions in the software industry focused on Internet-based visual collaboration tools. He is a fellow of the ASME, holds three U.S. patents, and is the recipient of numerous professional awards.

Dr. Brian Gilmore is a manager of Worksite Systems Engineering at the John Deere Moline Technology Innovation Center. While at John Deere, he has worked with several universities to develop advanced technologies for John Deere’s virtual reality applications. Before coming to John Deere, he was a professor of mechanical engineering and faculty research associate at The Pennsylvania State University. He is a graduate of Purdue University.

Jerry Dubien is a manager of University R&D Relations at the John Deere Moline Technology Innovation Center in Moline, IL. He is responsible for developing and sustaining strategic university relationships to support Deere’s global enterprise technology innovation needs. He also leads research personnel in advancing Deere & Company capabilities in user-interface design and immersive collaboration technologies. He received a Ph.D. in industrial engineering, specializing in bioenvironmental engineering, human factors and ergonomics from Kansas State University. His research and publication contributions have been primarily in the fields of human factors, immersive collaboration, and development of new system design tools.