Computer-Aided Design and Applications

http://www.cadanda.com

A Virtual Environment for Satellite Assembly

Arun Muthaiyan[1] and J. Cecil[2]

[1]New Mexico State University
[2]New Mexico State University, jcecil@nmsu.edu

ABSTRACT

This paper describes the design and implementation of a Virtual Environment to support process design activities related to satellite development. Such an environment enables the early identification of infeasible design ideas by involving process engineers (who are involved in downstream assembly, integration and other functional activities) in the satellite product design activity. A framework for the design and use of virtual prototyping environments is also outlined. The initial development of this Virtual Reality environment is to aid in the analysis of assembly related process issues focusing primarily on nanosatellites.

Keywords: virtual reality, satellite assembly, process design.
DOI: 10.3722/cadaps.2008.526-538

1. INTRODUCTION

The life-cycle activities involved in product development of satellites and space systems are complex and typically include design, planning, manufacturing, assembly, integration, testing, launch and operations. In recent years, there has been a greater emphasis by government and commercial organizations on the need to reduce the overall product development costs and the lead-time to launch space systems and vehicles [5,7,8]. Examples of space vehicles include satellites, space shuttles, telescopes, unmanned space probes and other larger systems such as the International Space Station. Virtual Prototyping is one of several approaches, which can facilitate better analysis and earlier identification of infeasible design ideas in the space vehicle development activity [3,5,7,8]. By enabling engineers involved in the product and product design activities to work in a cross functional manner (and addressing downstream problems early in the design cycle), virtual prototyping contributes towards the overall cost reduction and product quality improvement of space vehicles and systems. For ease of document preparation, authors are requested to type their papers directly into this file.

The need for life cycle integration and concurrent engineering has been emphasized in the space systems domain only in recent years [5, 7, 8, 10]. While there are numerous issues, which have contributed to this scenario, the primary reasons relate to an emphasis on developing innovative product designs with a lack of an adequate focus on process design issues. Space systems related research has traditionally focused on design innovation with the emphasis on launching high quality products within specified schedules (rather than reducing overall costs without sacrificing product quality)[3, 5].

The approach discussed in this paper was developed as part of a long term initiative related to the adoption of virtual engineering and distributed manufacturing principles in the space systems development domain. The motivation for the research and educational activities related to satellite process design was derived from two research initiatives: the first initiative titled 'ASTEC' (for Advanced Space Technology and Engineering Concepts) focused on the integrated life-cycle development of the manufacturing of one hundred nanosatellites. The second initiative was part of an exploratory activity aimed at studying the need and surrounding issues related to Affordable Space Systems [19]. Early evaluation of product designs from downstream perspectives (including manufacturing, assembly, testing and

launch) were identified as crucial issues, which needed to be addressed as part of long term efforts to reduce the lead time to develop new space systems of high quality at lower costs. Virtual Prototyping was one of several techniques identified as possessing the potential to enable cross functional engineers from various areas of specialization to propose, conceptualize, analyze, modify and compare product and process design alternatives. A Virtual Satellite Assembly (VSAT) environment was created to support evaluation and comparison of candidate satellite designs. In this paper, the primary focus is on the design and implementation of VSAT.

Virtual Prototyping based techniques have been explored for a variety of product and process design environments [1-17]. There are various categories of satellites including small satellites, nanosatellites and pico satellites. While there is no industry wide consensus on the specific definitions pertaining to these categories, the following classification scheme is recognized for purposes of this paper: Small satellites weigh less than hundred kilograms in weight, microsatellites are less than fifty kilograms, nanosatellites are typically less than ten kilograms and picosatellites weigh less than one kilogram.

2. VIRTUAL PROTOTYPING BASED APPROACH AND ENVIRONMENT

The long-term objective of this research is to create a Virtual Reality based prototyping environment, which would enable cross-functional teams composed of product engineers, and process specialists (in manufacturing, assembly, testing and launch) to work in a concurrent engineering (CE) based manner during product development. The short-term objective includes the ability to study, analyze and compare candidate satellite design alternatives from an assembly point of view. While the primary domain of interest is nanosatellites, the virtual satellite assembly environment built (VSAT) can be used to support process design of other classes of satellites as well.

An integrated approach has been developed for the creation of a virtual environment for satellite development. The distinguishing features of this approach include the use of an information oriented enterprise model which guides the development of the Virtual Assembly environment (VSAT) and the use of information modeling techniques, which was used as the basis to understand the satellite development activities target product and process related activities (as well as other activities from design through launch).

2.1 Using Information Models to Design the Virtual Satellite Assembly Environment (VSAT)

An important feature of the virtual prototyping framework is the virtual prototyping environment (VPE); the VPE is a generic term for any virtual prototyping environment. For the satellite analysis initiative, this VPE is termed the Virtual Satellite Assembly (VSAT) Environment. One of the activities in this research was the creation of an IOEM, which can be used to design and develop the virtual assembly environment. This was achieved using a language called the enterprise (Engineering) Modeling Language (eEML). This model was used as the basis by which various relevant tasks were accomplished; the relationships and constraints were modeled explicitly using functional and temporal precedence attributes. We refer to this model as an information oriented enterprise model (IOEM); such a model not only describes what activities are to be accomplished by team members but also how they should be accomplished. For a complex project, a software enterprise may be involved in developing virtual environments; the engineers, managers, software designers and programmers who would employ various resources collectively can be viewed as an 'enterprise'. In general, such IOEMs are useful in providing a formal foundation to identify information drivers (needed by the various engineers and software architects), the available computing and engineering resources to be used in achieving stated design and process objectives, the constraints imposed in accomplishing identified activities and the decision outcomes (which are expected as various activities are completed). IOEMs are versatile and can be used as the basis for planning, analysis and communication among team members involved in creating the VPE.
eEML has both a methodology and rules for creating the various diagrams. At the highest level, the activity to be modeled is termed as the Focus Unit, which appears as a verb representing the activity of interest. The Focus Unit (FU) can in turn be decomposed into activities or sub-tasks called as Task Units, which in turn can be described using lower level Function / Process (FP) Units. The information attributes used in an eEML model to represent and capture any information rich context include the Influencing Criteria (IC), the Associated Performing Agents (APA), the Decision Objects (DO) and the Task Effectors (TE). Using these attributes, a target set of activities (or a life cycle activity in manufacturing) can be modeled, studied and analyzed at various levels of abstraction. The highest level within a modeling context is the E-0 level (which corresponds to the Focus Unit level). Additional information on eEML and IOEMs can be found in [6].

A list of the major attributes used in eEML is shown in Figure 1 along with their abbreviations (which appear in the eEML diagrams). The top level eEML diagram created for building the VSAT environment is shown in Figure 1. Figure 2 is a decomposition of Figure 1; each of the boxes in Figure 2 was decomposed; however, for purposes of brevity, these diagrams have not been included in this paper. The E-0 level diagram (in Figure 1), which represents the highest level of abstraction, illustrates the general layout and use of attributes in eEML. The primary, secondary and tertiary importance of the various attributes is indicated using 'P', 'S' and 'T' symbols as in Figure 2. This is determined by the modeler based on their perception of the level of importance of a specific attribute. The numerical suffix (of an attribute in decomposition as in Figure 2) after the abbreviations (such as CO-0.1) refers to the numerical identifier ('1') of that attribute ('CO') in a parent level ('0', see Figure 2). For example, CO-N.1 refers to the first attribute in the category CO at the E-'N' level. A modeled (target) activity E-0 can be decomposed into a set of related activities E-1, E-2, etc (as shown in Figure 2). In general, before the creation of this eEML model, the project team drafted the context and modeling perspective and identified the project objectives formally. At this E-0 level, the various ICs, APAs, and DOs are identified as well as prioritized (as primary, secondary and tertiary); the temporal relationships among these activities are also captured using appropriate junctions. Subsequently, the tasks that comprise the focus unit are identified in the decomposition and the process is repeated.

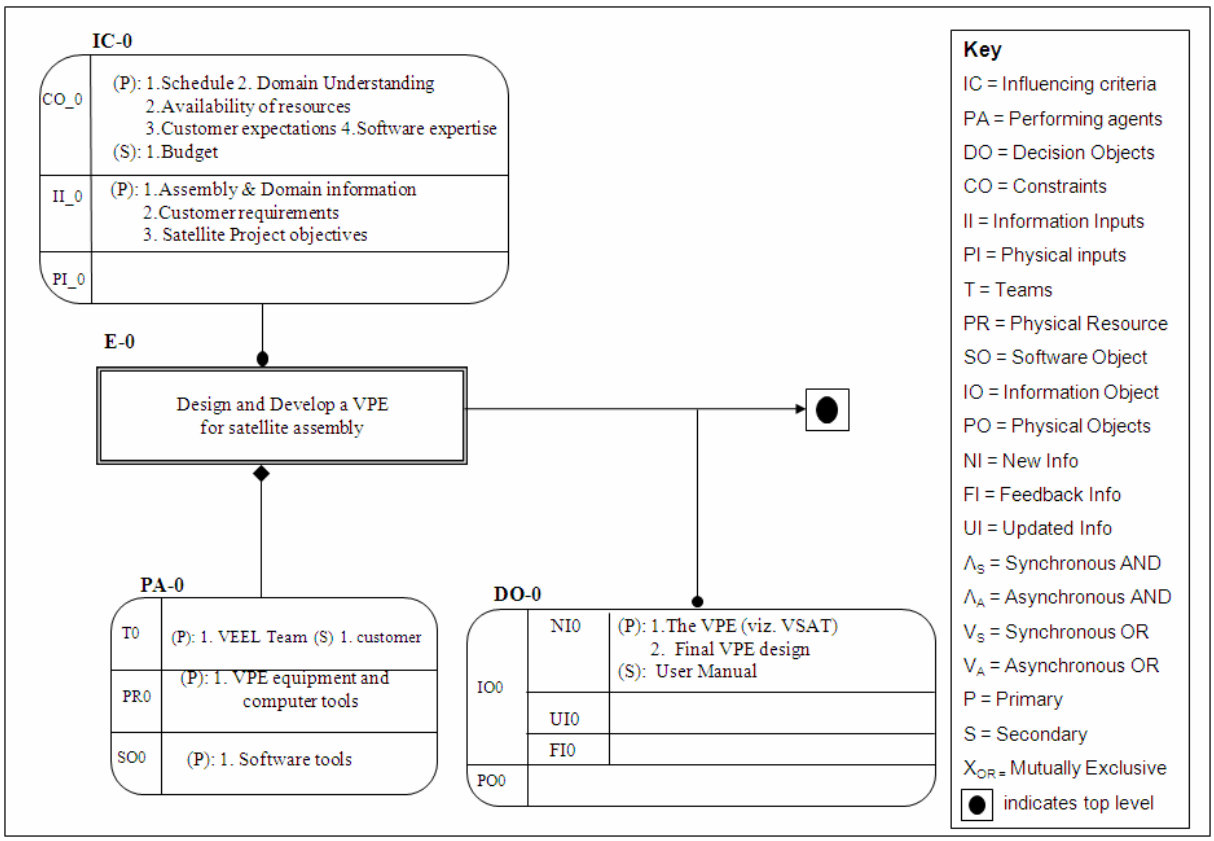

Fig. 1: The E-0 level diagram.

Figures 1 and 2 show the top-level model and its decomposition. Figure 1 focuses on the top-level context of the activity 'Develop a Virtual prototyping environment for micro assembly'. The influencing criteria are shown in Figure 1 and include constraints (such as budget, customer expectations, project schedule, etc.) and information inputs provided to the research team (such as information relating to the domain of micro assembly as well as customer requirements and project objectives). The key decision objects include the virtual prototyping environment (or VR system) and the VR system design.

The major phases involved include the following:

1. Understand customer requirements (E1)

2. Understand and study target domain (E2)
3. Design the virtual prototyping environment (or VPE) (E3)
4. Build the VPE (E4)
5. Validate the VPE (E5)

In this paper, the scope of discussion and model illustration is limited to the first four phases. Figure 2 shows the decomposition of the target activity in Figure 1. In Figure 2, the diagram shows an elided view of the activities and the temporal precedence relationships (indicated using the junction boxes). The various phases (E1 through E5) were completed based on the eEML models developed. In the following discussions, sometimes the past tense is used to highlight the various tasks completed by the project team. The project team involved both engineers and software programmers.

Understand Customer Requirements (E1)
The first task in the creation of the VSAT involves understanding the specific requirements and needs of the customer. The relationship of E1 with respect to the other phases is shown in Figure 2. Apart from meetings and discussions, E1 includes reviewing customer requirements documents (provided to the research team) as well as other activities such as viewing videos provided by customer related to process and process design activities related to the project. Subsequently, a customer product specifications document (which is more detailed and developed by the project team) is developed, which is used to guide the VPE design and implementation. This document formally describes the various features and capabilities of the VPE. As indicated, this document can be created using word processing software as well as tools such as MS Visio.

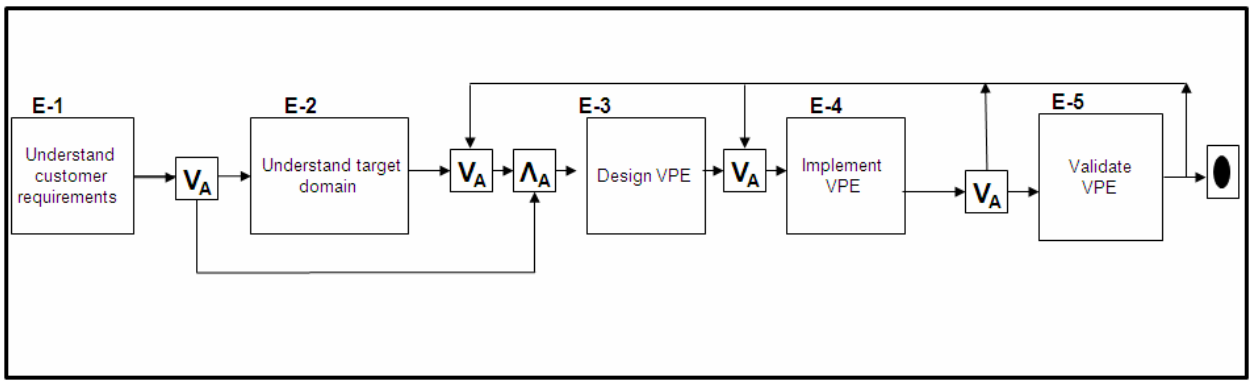

Fig. 2: Decomposition of the E-0 level diagram (elided view).

Understand target domain (E2)
Prior to building the virtual environment for satellite assembly, a crucial task involves understanding the target domain of interest viz. micro assembly. In a generic sense, it is possible that a project team may be familiar with a target domain; for this reason, the junction box (V_A) is used to indicate the asynchronous OR option in Figure 2 (that links E1 to E2 and E3). If the project team does not possess any background or understanding of the target processes in a target domain, the task of both designing and implementing a virtual reality system will be a significant challenge. For the creation of the VPE for satellite assembly, the project team became familiar with satellite systems, assembly steps, analysis tasks, etc; they also performed literature reviews through the web and traditional library mechanisms; numerous reports / papers were read and additional discussions were held with industry experts. Videos of satellite assembly activities were also viewed to obtain a better understanding of the satellite assembly domain. In the diagrams, the term 'VPE' refers to the Virtual Environment for Satellite Assembly.

Design the Virtual Environment (E3)
This activity is perhaps the most important of the various phases leading to the creation of the virtual environment. It can be decomposed to involve the following sub-activities: Develop a preliminary design of Virtual Environment, Discuss preliminary design with customer, and then subsequently design a detailed design of the Virtual Environment. The preliminary design activities include (a) identifying the various hardware, sensors, platform and programming

language options, (b) identifying the major software and analysis modules for implementation, (c) developing an initial block diagram representing the preliminary architectural details of the VPE, which are then subsequently formalized into a set of UML (Unified Modeling Language) models (including collaboration, sequence and object diagrams). A Collaboration diagram is an interaction diagram that emphasizes the structural organization of the software objects that send and receive messages. A sequence diagram is a model that describes how groups of software objects collaborate in some behavior over time and capturing the behavior of a single use case. It shows the objects and the messages that are passes between these objects in the use case. An example of a sequence diagram is shown in Figure 5. The dynamic interactions between the objects are showed by an arrow marked straight line with the suitable messages. The task 'Develop detailed design of virtual environment' includes reviewing preliminary design (E331), followed by creating class diagrams and design documents for the various modules identified (such as user interface, assembly module, etc.). Subsequently, the collaboration diagrams detailed the information exchange among the various modules of VSAT are developed. This is followed by the detailed algorithm development for the various analysis and simulation tasks to be supported by VSAT. An example of an algorithm used is the genetic algorithm based approach discussed in the next section of this paper. All of these together are submitted as a final design document.

Build the Virtual Environment (E4)
This activity primarily identified the various software programming and module implementation tasks for the various modules in the target VPE. These included using the detailed design of the virtual environment (modeled in E3) and creating the various modules in the VSAT environment (as shown in Figure 3) including user interface, the collision detection modules, the core assembly and analysis modules, the immersive interface modules. Virtual reality equipment and devices such as motion trackers, stereo eyewear were selected by the team to complete the development of the VSAT environment.

Validate the Virtual Environment (E5)
This activity involves using experts from the satellite industry interact with the VSAT environment, perform analysis and validate the assembly analysis environment. This included identifying several test case scenarios, obtaining feedback from the experts, and incorporating any identified modifications to the VSAT environment.

The eEML models were built in an iterative manner and involved a group of four researchers. Due to page limitations, not all the decompositions have been shown in this paper. A detailed glossary of the activities and the modeled attributes was also developed.

2.2 Understanding Satellite Development Activities

For the creation of the simulated VR environment (which forms the core of the VPE), there is a need to use structured methods, which can capture the intricate details of a target set of processes and activities. For the described research, IOEMs were created and used as a communication basis to ensure a better understanding of target activities. For example, project team members not familiar with a certain activity (such as satellite assembly) could become familiar with that activity by either physically observing this activity, viewing video tapes and / or studying an information rich models which capture the essence of a target process. In this project, the emphasis was on the last option, which involves use of an IOEM. The IOEMs were created initially using eEML as well.

2.3 The VSAT Architecture

Based on the eEML model developed, the Virtual Satellite Environment (VSAT) was designed and developed. The architecture of VSAT implemented is shown in Figure 3. The major components of VSAT include the following:

1. User Interface Module
2. The Visualization Manager
3. Assembly Manager
4. Assembly Generator
5. Collision Detection module
6. Ancillary libraries and part file repositories
7. The Process Manager

The major architectural elements of VSAT are shown in Figure 3. The Process Manager's function is to coordinate the overall interactive process between the user and the VSAT environment. A screen view of the virtual environment is

shown in Figure 4. In Figure 5, the role of the Process Manager and the overall flow of events is shown is indicated using a sequence diagram. The Assembly Manager is responsible for coordinating the simulation of target assembly parts. The Immersive Part Handling Module is a software module provides the functionality that allows the user to pick up an object in the semi-immersive environment using WorkWand ® (in Figure 8). The User Interface Module provides a menu driven interface, which allows the users to select various analysis and interaction options within the VSAT environment. Collision Detection Module detects potential collisions between moving components during satellite assembly (both manual and automated modes). Some of the other routing functions such as loading part files onto the Scene graph and parsing inputed sequences are handled by other modules such as the Part Loading Module and the Parser Module respectively. Additional discussion of these modules follows.

2.3.1 User Interface Module

This module is responsible for allowing users to interact with VSAT. Users can interact using keyboard, mouse, and a WorkWand ® (from Fakespace). The movement and positions of the user in a physical world is tracked using motion sensors. Stereovision supporting eye-wear can be used to visualize, interact and navigate through the virtual environment. The main interface screen provides options to select a target world or environment, perform analysis, interact to study 'what if' options as well as allow the user to visualize target assembly / disassembly sequences using automatic or manual mode. This module primarily interacts with a Graphical User Interface (GUI) manager, an object interaction manager (OIM) and a visualization manager. The OIM allows users to virtually pick a target object and position them as desired. The primary use of this function is to allow experienced satellite manufacturing engineers to study various assembly alternatives and compare them to automatically generated sequences or other alternatives. Future uses of this capability would allow test and integration engineers to also perform other downstream engineering activities early in the satellite design process. The GUI manager (which has been implemented using X Windows) also allows users to switch from the manual assembly mode to an automated mode by selecting appropriate buttons on the work wand. Prior to beginning the analysis, users can use the GUI to bring in various part designs and move parts using translation and rotation commands. A virtual pointer is used to represent the user's position and orientation in the virtual environment (this can be seen in Figure 4, which is a screen view of the immersive environment).

The Object Interaction Manager uses various functions to enable 'immersive' assembly tasks. These functions include detecting if a user is close enough to a target object prior to allowing the user to 'pick' a part and constantly monitoring if specific buttons on the WorkWand has been selected (to indicate user's interest in picking a part or visualizing a specific sequence). The OIM also allows users to visualize disassembly sequences by using appropriate buttons on the WorkWand.

2.3.2 Visualization Manager

The simulations and representations in the Virtual Satellite environment are controlled through a visualization manager (which in turn interacts with the user through a user interface module). The visualization manager interacts also with the analysis modules which include an assembly path planner (with manual and automatic generation of assembly options) and a collision detection module. The various visualization activities have been implemented using Performer ® libraries.

The visualization manager has access to various functions such as loading and positioning / orienting a target set of parts (or objects) as well as providing responses depending on appropriate navigational commands received from the GUI or Object Interaction Manager. The visualization manager also interacts with other software modules that perform hierarchy management, which include parsing relevant hierarchy files (associated with a target assembly or world). Figure 5 show the sequence diagram for the VSAT system. This sequence diagram was also used as part of the VSAT design process (prior to building VSAT) to represent the spatial organization and interactions between the software objects, to visualize the sequence of the flow of control, and to show the concurrent processes and activities. The analysis activities include comparing alternate designs, performing 'immersive' assembly tasks and assessing design changes.

2.3.3 Assembly Generation Manager

The assembly manager controls the two basic assembly modes available in VSAT: manual and automatic. It works closely with the visualization manager, the GUI manager, and the object interaction manager. The overall representation and creation of a target assembly scene is communicated to the visualization manager by the Assembly Manager. If in automatic mode, the assembly manager provides the generated assembly sequence to the visualization

manager. It interacts with the collision detection module dynamically to detect collisions during a given assembly or disassembly. Depending on the user selection of buttons on the WorkWand, either the manual option of interacting with the virtual environment or viewing of a previously generated assembly / disassembly sequence is provided. To support these functional activities, the Assembly Manager works closely with the Object Interaction Manager. Scene creation, updating and deletion are achieved through interaction with the Visualization Manager. The GUI manager allows the user to exit or load a new assembly as needed.

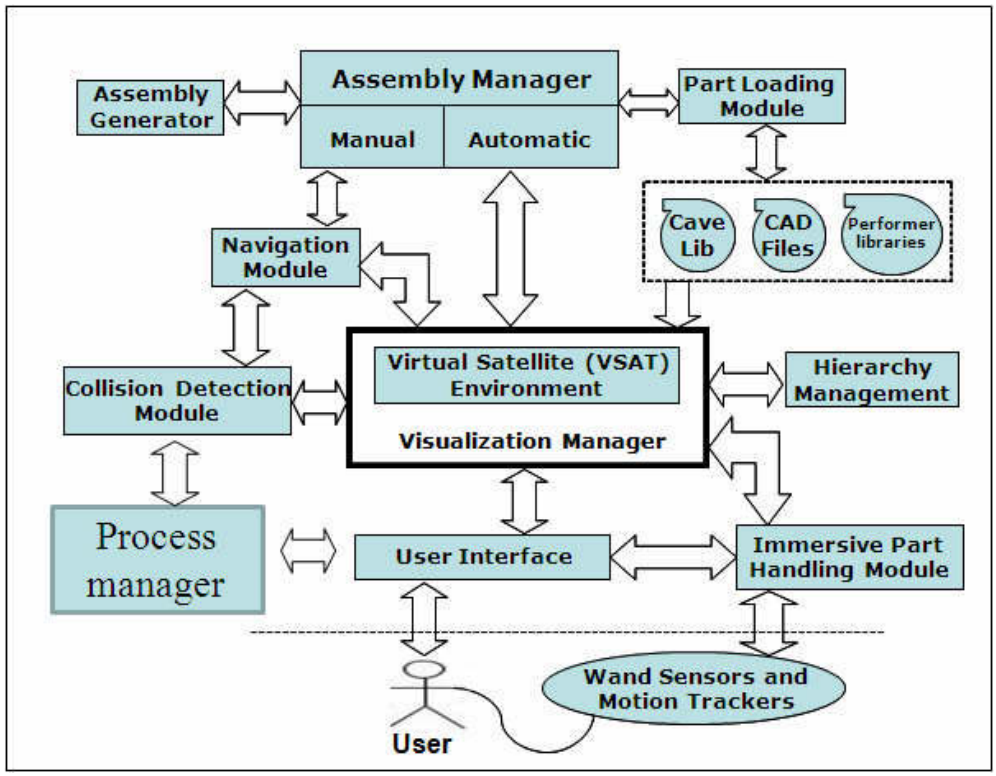

Fig. 3: The Modules in the Virtual Satellite Environment (VSAT).

Users have two options to interact with the virtual environment (a) manual and (b) automatic mode. Users can switch from one mode to another by depressing buttons in a WorkWand linked to the VR system (shown in Figure 8). In the Manual mode, users can manually interact with parts or objects in a given scene in the VR environment. In addition to moving, starting and stopping the system, experts can propose assembly alternatives, study potential problems and visualize impact of assembly decisions, proposed designs and the usefulness of existing resources. By selecting an appropriate button or a combination of buttons on the WorkWand, parts in the virtual environment can be picked, moved and placed at various locations. A virtual pointer helps in the guidance of task accomplishments by providing cues relating to the direction of motion and visualization.

In the Automatic mode, user interaction is limited. Based on the assembly constraints and the position of the various parts in an assembly environment, an overall assembly sequence can be generated (which is described in the next section). In addition, a file specifying a given sequence can also be specified through the user interface. In the automatic mode, the virtual environment only allows the user to navigate through the virtual environment, as well as provide options for start, pause, and stopping the simulation for a given assembly sequence (users cannot pick up a part when a simulation in automatic mode is being executed). The data or path coordinates are stored in a "Satellite.world" file, which maintains references to the CAD files of the various objects of interest (OOI) in a given assembly analysis scenario, information relating to the initial position and orientation, the final position and orientation, and the speed of the motion for each OOI.

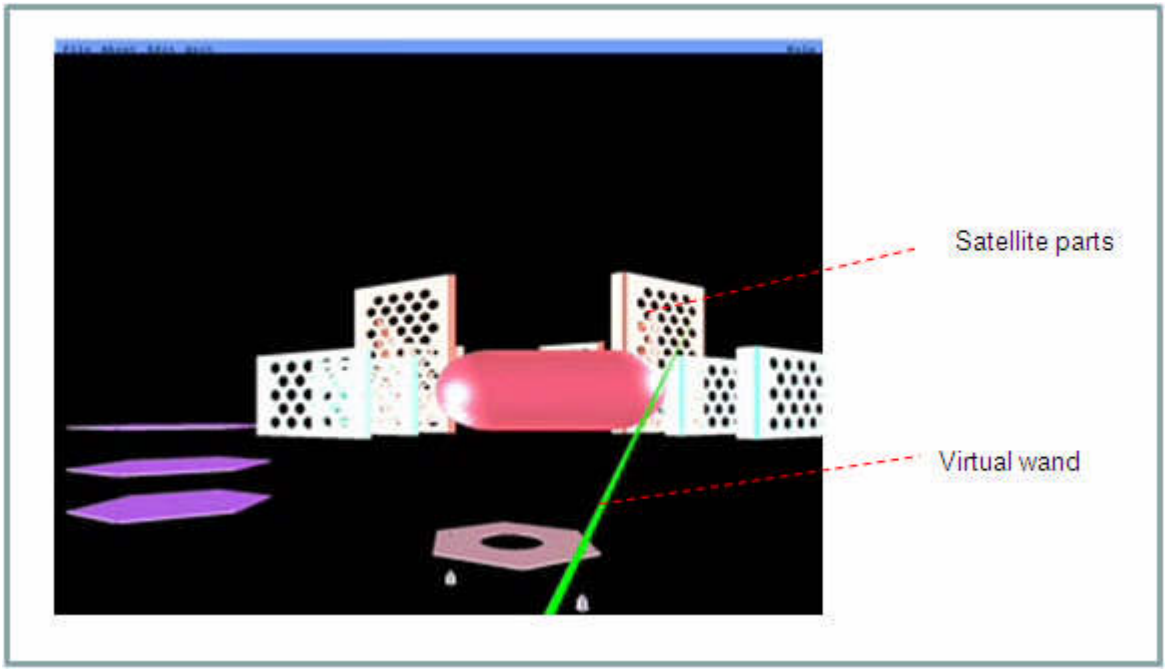

Fig. 4: A view of the Virtual Satellite Environment (VSAT).

The data structure format in the "Satellite.world" is described below:
Number <n> specifies the total number of objects in the system (n is the integer value from 1, 2, 3, …)
begin_obj refers to the beginning of the object's motion. The format of begin_obj command is:
{begin_obj <object name> <initial position x, y, z> <initial orientation x, y, z> <object scale in x, y, z> <final position x, y, z> <number of movement interval> <final orientation> <termination logic>}
The object <*.stla> is a keyword specifying the name of design file (or object of interest) which is an active component of a given simulation scenario. In general, a variety of CAD data formats can be supported; in this example, stla refers to stl data format files. Other file formats supported include *.vrml and *.obj files formats.
End_obj indicates that the motion of current set of objects is complete.
This data structure was originally proposed in [20] and has been modified to support multiple assembly sequences and motion segments.

Consider the "satellite.world" file (in Figure 7); it can be interpreted that the target assembly contains 28 objects; the first object name is BASE_DECK1 (which represent the virtual prototype of the BASE_DECK object using a stla file format) which has a start position of (-13.025, -43.955, -5) with the initial orientation (yaw = 90°, pitch = 0°, and roll = 0°). This object will move to a new position (-13.025, -43.955, 10) and new orientation (yaw = 90°, pitch = 0°, and roll = 0°) using 5 intervals. The scale of the appearance of the object will be the same as original file.

2.3.4 Assembly Generation
The sequence to complete a nanosatellite assembly has to be generated taking into consideration the initial and final positions of the various parts of a target assembly and presence of potential precedence constraints involving a target set of objects. A genetic algorithm based approach has been developed to determine the assembly sequence involving various assembly parts. The algorithm can be used to assemble the overall satellite or focus on only the assembly of a specific module (for example, a satellite design may be composed of the propulsion, command and cold gas module). Cross over, mutation and inversion operators were used to generate new child links. The objective function is the time consumed in completing a target satellite assembly. The role of the GA based generator is indicated in Figure 3 using the term 'assembly generator'. This module generates near optimal assembly sequences.

In a satellite with 'n' components, the assembly sequence can be represented as a list $S_1, S_2, \ldots S_j$ (where S_j refers to a component to be placed). Beginning with a randomly generated initial set of assembly sequences, 3 types of genetic operators can be applied to generate children or offspring links. At each iteration, a certain number of links are chosen based on an evaluation function (which is the distance traveled by an assembly robot in completing the assembly task). At each iteration the links with the lowest distance are chosen as new parent links. After each iteration the distance travelled will continue to decrease. The iteration comes to an end when there are no substantial improvements in the distances calculated or a specified number of iterations are reached. For the given problem, three classes of genetic operators including cross over operator, segment reversal operator and mutation operator are used to generate new child links or offspring sequences. A partial section of the linked list (which represents the placement sequence) is referred to as the Segment.

Reversal Operators

Segments can be reversed (using the segment reversal operator) and transported (using the cross over operator) to generate these new states or child links. When this operator is applied, a segment of a sequence is selected and the order is reversed. This operator is used to generate 15 % of the new child links.

Crossover Operator

This operator can be applied on two parent links S_1 and S_2, which results in a new child link. A cut point can be randomly chosen corresponding to a position k in the two links. Then, the first portion of the first link L_1 and the second portion of the second link are merged to form the new child link S. Care is taken to ensure that no components or genes are repeated or missing in the new child link. When a component is repeated, the candidate gene next in the sequence (which is not present already in the sequence) is selected. The cross over operator is used to generate 70 % of the new child links.

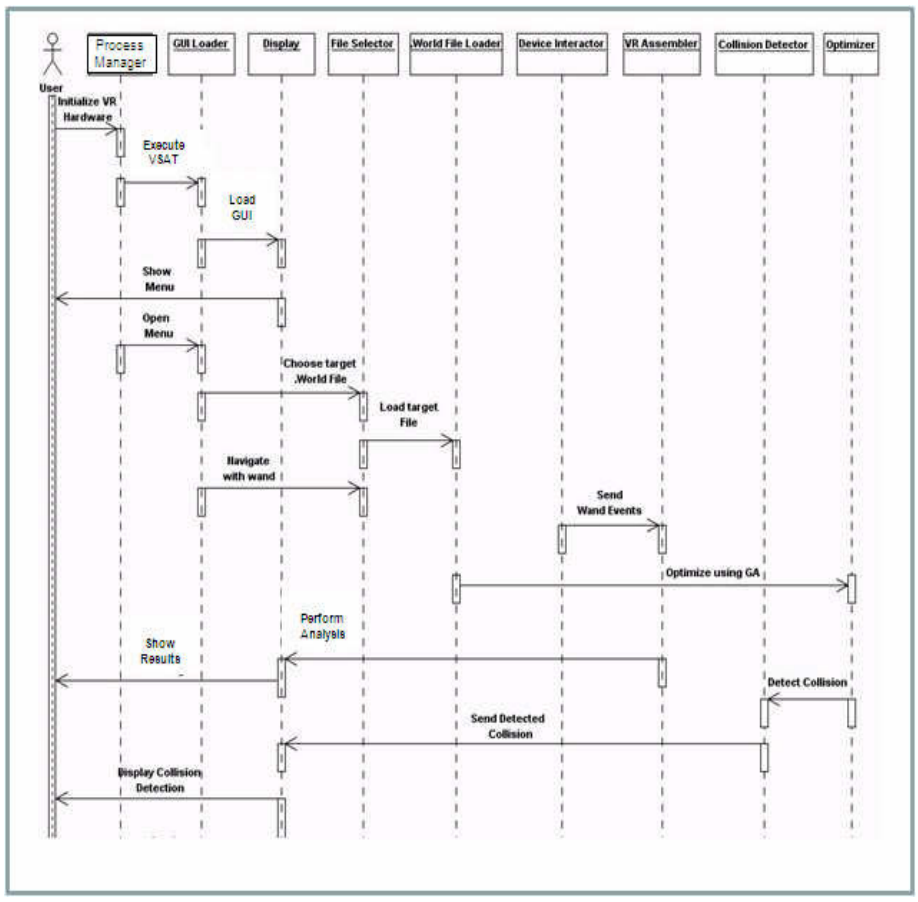

Fig. 5: The sequence diagram indicating the overall flow of events in VSAT.

Mutation Operator
In this operation, a single gene switches position to create a new child link (or sequence). Mutation adds to the diversity of a population and thereby increases the likelihood that the algorithm will generate individuals with better fitness values. This operator is used to generate 15 % of the new child links.

The major steps in the genetic algorithm based approach are as follows:

1. Generate candidate sequences randomly and form the initial parent links
2. Obtain the precedence constraints as a sequence for various parts in an assembly. For example, part A (which could be a deck plate) can only be inserted after another part B (which could be a cold gas thruster).
3. Generate child links using cross over and inversion operators. Ensure validity of child links by checking to see a part number appears only once in a sequence and that the precedence constraints are satisfied.
4. Select the child links SL_i with the lowest objective function values.
5. Repeat the process of generating new child links SL_{i+1} based on the new parent links SL_i

The process is stopped when the objective function value shows no significant decrease.

```
number 28
begin_obj BASE_DECK1 -13.025 -43.955 -5 90 0 0 1 1 1 -13.025 -43.955 10 5 90 0 0 -10000.0 -10000.0 -10000.0
      object BASE_DECK.stla
end_obj
begin_obj HSHCSC1 -14.75 -43.5 5 90 0 0 1 12.3 1 -14.75 -43.5 103.0 5 90 0 0 -10000.0 -10000.0 -10000.0
      object HSHCSC.stla
end_obj
begin_obj HSHCSC2 -7.75 -39.45 5 90 0 0 1 12.3 1 -7.75 -39.45 103.0 5 90 0 0 -10000.0 -10000.0 -10000.0
      object HSHCSC.stla
end_obj
begin_obj HSHCSC3 -0.75 -43.5 5 90 0 0 1 12.3 1 -0.75 -43.5 103.0 5 90 0 0 -10000.0 -10000.0 -10000.0
      object HSHCSC.stla
end_obj
begin_obj HSHCSC4 -0.75 -51.6 5 90 0 0 1 12.3 1 -0.75 -51.6 103.0 5 90 0 0 -10000.0 -10000.0 -10000.0
      object HSHCSC.stla
end_obj
begin_obj HSHCSC5 -7.75 -55.65 5 90 0 0 1 12.3 1 -7.75 -55.65 103.0 5 90 0 0 -10000.0 -10000.0 -10000.0
      object HSHCSC.stla
end_obj
.....
.....
```

Fig. 6: A sample 'world' file

2.3.5 Collision Detection Module
The collision detection module detects potential collisions between moving components during satellite assembly (both manual and automated modes). In this approach, an enclosed box, is first generated, which fits around the Primary Object of Interest (POI). Similarly, enclosed box are created around another desired Object of Interest (AOI) which may be close to the POI; this AOI can be a Fixture, a tool or another component / satellite part. Subsequently, tests for intersection between the POI and AOI's are conducted where each edge in POI is tested for intersection with each edge of AOI. This is repeated to detect collision between potential satellite parts during virtual assembly scenarios. If a collision is detected, the point at which collision occurs is highlighted to the user.

The user can manually attempt to assemble the satellite or can use the automatic mode. The collision detection module provides feedback during the assembly. The WorkWand (Figure 7) helps users to pick and place satellite parts manually as well as helps guide the navigation functions such as zooming and moving sideways, etc.

Other issues which can be studied include analysis of assembly options with parts which require less or no fixturing, ability of assembly tools to access and complete a specified task, etc. Using VSAT, concurrent engineering team members from design, planning, manufacturing and other areas can work together at proposing assembly alternatives and then studying what kinds of problems may be experienced. These teams can either manually input their assembly plans (or allow the assembly planner to generate them automatically) for the alternative designs and compare their assembly feasibility. Further, the assembly cell layout can be changed virtually as well to facilitate the study of innovative assembly solutions proposed by specific team members as well.

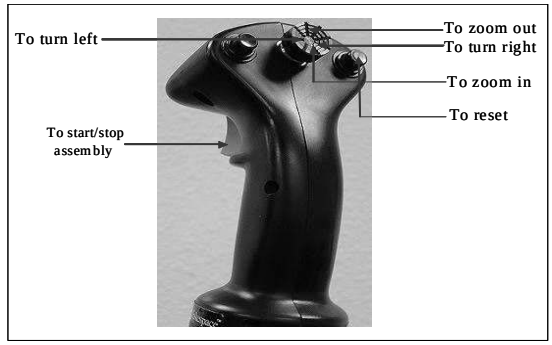

Fig. 7: WorkWand® used for interaction in the VSAT assembly environment.

3. IMPLEMENTATION AND DISCUSSION

The initial VR environment (which was referred to as VSA) was created on a Windows based platform. Currently, VSAT is implemented on a Silicon Graphics (SGI) Octane workstation using C++ and Performer libraries. The VR equipment include with stereographic eye-wear (from Stereovision), a WorkWand ®, and Flock of birds ® sensors (from Ascension) for motion tracking and control. Users can navigate 360 degrees vertically, horizontally, or both through the virtual environment using wand joystick and buttons on the WorkWand, which enables users to experience 'immersion'.

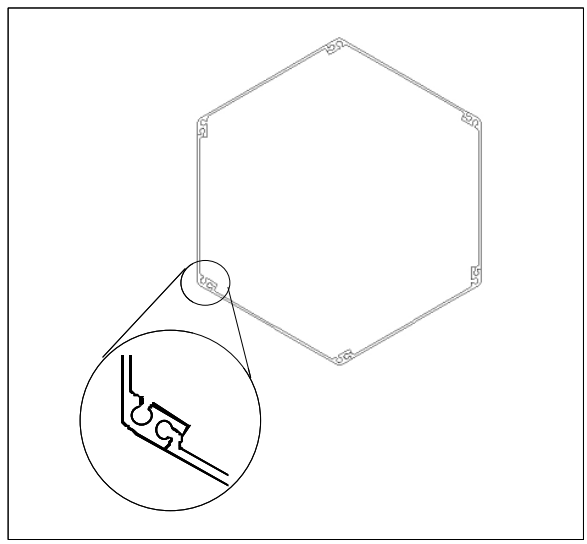

Fig. 8: Using self-indexing parts to simplify assembly processes.

One of the satellite designs which was studied and modified using VSAT had 3 main modules: command module, cold gas module and propulsion module. Using VSAT, candidate designs were proposed and modified virtually. For example, one of the design alternatives proposed in VSAT was of a modular construction; this design is more process friendly as the modules can be simultaneously constructed because of their 'modular' design and pre-defined interfaces. Such a design minimizes the impact of revisions: when changes to the propulsion module design have to be adopted, the product design and assembly of the command module can remain the same. Other process team members including subcontractors were able to study potential alternatives to complete their assigned sub-assemblies (or assemblies) as well as study the various interface options to complete the overall satellite integration. Figure 8 provides an illustration of a simplified structural design of a satellite to reduce assembly costs and time. The outer satellite structural shell has self-indexing parts and sub-assemblies. With only one way to assemble, this design

minimizes the possibility of assembly errors as well as reduces the assembly time involved by using interlocking side panels. Additional designs issues such as integration of the satellite and the launch bus for various space program initiatives (such as TechSat 21) were also proposed and compared using VSAT.

VSAT is also being used to support teaching of virtual engineering concepts to university and school students. Part designs relating to design and assembly of satellites were obtained from industry partners who helped develop laboratory and lecture modules related to this subject; undergraduate students were also introduced to research topics such as integration of satellites onto a launch bus, comparison of assembly alternatives and the importance of virtual environments in facilitating better communication among engineering team members. The research and curriculum development efforts related to satellite assembly and product development are part of a broad based initiative entitled 'DHARMAM', which focuses on emerging Information Technology (IT) based areas such as virtual enterprise engineering, information based manufacturing, distributed collaboration, virtual prototyping and electronic commerce [21].

4. CONCLUSION

This paper discussed the need for virtual prototyping in the satellite development domain and provided a description of the Virtual Satellite (VSAT) environment, which was created to support virtual prototyping of assembly alternatives. The key aspects of VSAT's architecture include a visualization manager, an assembly manager, a user interface module and a collision detection module. The VSAT environment is also being used by undergraduate and graduate students to study designs and compare alternatives as part of new course modules related to virtual engineering and distributed manufacturing. Future work involves using VSAT for undergraduate students involved in senior design projects to compare candidate designs and simulate various assembly tasks. The user interface is currently being modified to provide additional interactive capabilities and cues [21].

5. ACKNOWLEDGEMENT

Funding from the National Science Foundation (NSF CRCD Grant Number 0196303) and Air Force Research Laboratory (AFRL, Albuquerque) is gratefully acknowledged.

6. REFERENCES

[1] Banerjee, A.; Banerjee, P.; Dech, F.; Ye, N.: Assembly planning effectiveness using virtual reality, PRESENCE - Teleoperators & Virtual Environments, 8(2), 1999, 204-217.
[2] Banerjee, A.; Banerjee, P.: A Behavioral Scene Graph for Rule Enforcement in Interactive Virtual Assembly Sequence Planning, Computers in Industry, 42(2-3), 2000, 147-157.
[3] Cecil, J.: DHARMAM: An University – Industry Partnership to enhance Manufacturing Engineering Education, Proceedings of the Advanced Manufacturing Institute's International Conference on University & Manufacturing Industry Collaboration, August 12-13, 2002, Kansas City, Missouri.
[4] Cecil, J. A.; Kanchanapiboon, A.; Kanda, P.; A. Muthaiyan: A Virtual Prototyping Test bed for Electronics Assembly, Proceedings of the IEMT / IEEE symposium, San Jose, July 2002.
[5] Cecil, J. A.: Advanced Space Technology and Engineering Concepts (ASTEC) report, Utah State University, October 2000.
[6] Draper, J. M.; Dessouky, M. M.; Verma, S.; Bailey, D. E.; Rickel, J.: A methodology for developing a web-based factory simulator for manufacturing education, IIE Transactions, 33 (3), 2001, 167-180.
[7] Goldin, D.; Venneri, S.; Noor, A.: New Frontiers in Engineering, Mechanical Engineering, 120(2), 1998, 63-69.
[8] Goldin, D.; Venneri, S.; Noor, A.: Ready For the Future?, Mechanical Engineering, 121(11), 1999, 61-70.
[9] Korves, B.; Loftus, M.: Designing an immersive virtual reality interface for layout planning, Journal of Materials Processing Technology, 107(1-3), 2000, 425-430.
[10] Noor, A.; Venneri, S.; Housner, J.; Peterson, J.: A Virtual Environment for Intelligent Design, Aerospace America, 119(4), 1997, 28-35.
[11] Srinivasan, H.; Figueroa, R.; Gadh, R.: Selective disassembly for virtual prototyping as applied to de-manufacturing, Robotics and Computer-Integrated Manufacturing, 15 (1), 1999, 231-245.
[12] Ye, N.; Banerjee, A.; Banerjee, P.; Dech, F.: Assembly Planning in Traditional and Virtual Environments, IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews, 29 (4), 1999, 546-555.
[13] Angster, S. R.: VEDAM: Virtual Environments for Design And Manufacturing, Ph.D. Dissertation, Washington State University, Pullman, WA, 1996.
[14] Banerjee, P.; Kesavadas, T. (editors): Industrial Virtual Reality: Manufacturing and Design Tool for the Next

Millennium, Proceedings of the NIST – ASME Industrial VR Symposium, University of Illinois – Chicago campus, Chicago, IL, Nov.1-2, 1999.

[15] Barrus, J. W.: The Virtual Workshop: A Simulated Environment for Mechanical Design, Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA, September 1993.

[16] Chuter, C.; Jernigan, S. R.; Barber, K. S.: A Virtual Environment Simulator for Reactive Manufacturing Schedules, Proceedings of the Symposium on Virtual Reality in Manufacturing Research and Education, University of Illinois, Chicago, 1996.

[17] Deitz, D.: Modeling a Virtual Ocean, Mechanical Engineering, May 1998, 66-68.

[18] Cecil, J.; Punal, M.: Exploring the role of Information Modeling in the Design and Development of Internet based Systems, Proceedings of the Annual Industrial Engineering Research (IERC) conference, Portland, OR, May 16-17, 2003.

[19] Cecil, J.: Researching the creation of a Center for Affordable Space Systems, Project Report, Virtual Enterprise Engineering Laboratory (VEEL), New Mexico State University, Las Cruces, NM, October 2001.

[20] Banerjee, A.: Creation of a virtual robotic cell, Project Report, Industrial Engineering Department, Texas A&M University, College Station, TX, Spring 2001.

[21] Cecil, J.: Using Virtual Environments to assess candidate satellite designs, Project Report, Virtual Enterprise Engineering Laboratory (VEEL), New Mexico State University, Las Cruces, NM, October 2007.