



OPEN ARCHITECTURE FRAMEWORK FOR INTEGRATED VIRTUAL PRODUCT DEVELOPMENT SYSTEMS

Scott Angster and Sankar Jayaram [{{bios}}](#)
Washington State University
Pullman, Washington, USA

Editor: Mel Siegel

ABSTRACT

In the current marketplace, products are required to proceed rapidly from conceptualization to production. This has put a great deal of pressure on the current state of computer software systems. These include computer-aided design and manufacturing systems, design for assembly systems, design for manufacture systems, and manufacturing simulation systems. Modern product development processes call for rapid designs and adapting designs to suit ever changing customer requirements. Virtual prototyping is allowing engineers to quickly create digital prototypes, allowing for quick evaluation of conceptual designs. New technologies such as virtual reality are now being used to aid engineers in the area of virtual prototyping. Existing systems using virtual reality technology are limited in their expandability, customization, or usability with current design software systems. This paper discusses an expandable and customizable architecture aimed at integrating virtual product development tools.

1. Introduction

The current marketplace has demanded that companies reduce the time and cost involved in taking a product from conceptualization to production. Software for computer-aided design, computer-aided manufacturing, design for assembly, design for manufacture, and manufacturing simulation have assisted in this reduction of time and cost. However, there is still often a need to

produce physical mockups to test assembly requirements, manufacturing plans or ergonomic functionality. This has led to the use of new technologies such as virtual reality to further aid engineers in solving these problems [9],[12].

Current CAD/ DFA, DFM and manufacturing simulation tools provide the designer with valued CAM, information, but fall short of providing the information that virtual reality technology can provide. CAD/CAM software provide powerful design environments using parametric design methods and solid modeling. Current DFA software attempt to reduce the number of parts by merging several parts into one. This may, in fact, cause handling difficulties later on in the manufacturing process. DFM software may indicate that the part can be produced using a series of manufacturing processes without the knowledge of the actual manufacturing plant's capabilities. Current manufacturing simulation software are often limited to predefined functionality for both machines and human models. If the projected plans for manufacturing a part involves a human operator or handler, a predefined human model can not give back "true human feedback" concerning ease of fit, ease of handling, etc.

The modern product development process calls for rapid design through manufacturing cycles, agile manufacturing systems, adapting designs to suit rapidly changing customer requirements and preferences, use of centralized advanced manufacturing facilities, and outsourcing fabrication. The current suite of product development and rapid prototyping tools are not geared for these new scenarios. Virtual prototyping is a new concept which allows designers to create digital prototypes and evaluate the products thoroughly before a physical prototype is created at a remote location or by a subcontractor. This significantly reduces time to market and increases the competitiveness of a company. However, the software tools used for virtual prototyping (from traditional CAD systems to virtual reality based manufacturing simulations) are so varied in their architecture that it is almost impossible to create a cohesive set of virtual prototyping tools for use by any organization. This is compounded by the fact that customers and subcontractors have completely different sets of tools for performing the same tasks.

Many of the issues in virtual prototyping are addressed by virtual reality technology. The ability to view full scale, three-dimensional models can increase the designers productivity by allowing a more natural analysis of a model. Virtual flythroughs of manufacturing simulation can provide a better understanding of the layout of a manufacturing plant. Virtual manufacturing is used to test and modify numerical control codes offline and to train workers. Virtual assembly will allow engineers to study ease of assembly and ease of handling. Virtual disassembly can be used to study repair and recycling issues. However, many of today's systems that have been built to exploit the power of virtual reality are limited in their expandability, customization or usability with

current design software systems. This paper discusses an expandable and customizable architecture aimed at integrating virtual product development tools.

2. Literature Review

Several groups have developed systems utilizing virtual reality techniques for early design decisions through the use of virtual flythroughs, virtual design, virtual assembly and manufacturing simulation. Washington State University has developed a system for the early design evaluation of automobile interiors. This system utilizes Pro/Engineer models that are brought directly into a virtual design environment. Once immersed in the virtual environment, a user can evaluate the design, evaluate alternate designs and conduct ergonomic studies using full human body tracking. This work was a continuation of a feasibility study that provided successful results in the use of virtual reality for design [3],[4].

Through joint work at the University of Illinois, Chicago, and Purdue University, a prototype virtual reality based computer-aided design system has been designed and implemented. The focus of this work is to allow a simplified method of designing complex mechanical parts through the use of virtual reality techniques. Interaction with the environment is handled through the use of a wand, keyboard and voice recognition [18].

Research at the Georgia Institute of Technology is focusing on early design changes based on demanufacturing and servicing criteria. A Virtual Design Studio is being developed to enable designers to interact with recycling and tooling experts in a virtual environment. Parts that are being designed will be disassembled within the virtual environment to identify and correct demanufacturing and servicing problems [17].

The University of Bath in Bath, UK has developed an interactive virtual manufacturing environment. This system models a machine shop floor containing a three axis numerical control milling machine and a five axis robot for painting. The user can mount a workpiece on the milling machine, choose a tool, and perform direct machining operations, such as axial movements or predefined sequences, or load a part program from memory [8]. The system uses stereoscopic glasses to produce the 3D environment for the user, and a mouse is used as the primary input device. This software does not provide the users with the ability to create their own machines, or to interact with them in a natural manner. All interaction between the user and the environment is through the use of a mouse. Also, the software is not linked to any integrated, parametric CAD/CAM system.

A virtual workshop for mechanical design was developed at Massachusetts Institute of Technology [7]. The goal of the project was to develop a simulated workshop for designers to do conceptual design work while having to take into account manufacturing processes. The simulated workshop consists of a band saw, a drill press, a milling machine, a radial arm saw, and a table saw. Once several parts have been made, they can be assembled together. This software provides only a two-dimensional interface to the user. All interaction between the user and the environment is done via the mouse. The user is also limited to those machines provided by the software. There is no link to an integrated, parametric CAD/CAM system.

Deneb Robotics has commercially available software for manufacturing simulation, virtual milling, virtual spray painting, virtual arc welding, and telerobotics. Most of these systems are precompiled software tools where all work is done using Deneb's graphical user interface on the screen for setting up the manufacturing plant, etc., and all subsequent interaction is also done on the screen via the mouse and keyboard. Deneb has developed an immersive system that allows the users to view and interact with their simulated environments using a head mounted display and other virtual reality hardware [21].

Technomatix Technologies has developed several products in the area of virtual manufacturing. Robcad™ has been developed as a computer-aided production engineering tool for the analysis of robotic applications in a virtual environment. These include welding, laser cutting, and painting. Robcad also allows for offline robot programming and an open system architecture for developing user specific features and applications. Another product, Part, has been developed for computer-aided numerical control process planning and programming. Part provides machine tools, machine setup, machining methods, and cutting tools. Part will automatically create NC programs and process plans given computer-aided design models [23].

Prosolvia Clarus AB manufacturing has created a virtual manufacturing toolkit that is adapted for Technomatix Robcad modules. Users can interactively visualize a manufacturing simulation through the use of virtual walkthroughs of detailed computer-aided design data. This toolkit features an open architecture for users to create their own applications. Clarus is also researching and developing virtual assembly tools [20].

Resolution Technologies has created Virtual Mockup™, a system for flythrough analysis of computer-aided design models. Virtual Mockup provides a set of tools to the user that allows the user to such things as view the models in real time, query the models data set, remove objects from the model, and save viewing motion paths for playback [22].

Angster [2], Angster, et. al. [6], and Narayanan [15], created a device independent, object oriented, knowledge based system framework, called the Expert Consultation Environment. This framework gives programmers of computer-aided design applications the ability to add knowledge based systems to their applications without having to worry about the details of knowledge based systems programming. This allows the programmer to concentrate on knowledge acquisition for the system. A prototype expert system was developed for parametric, multidisciplinary, conceptual design of general aviation aircraft.

Jayaram, et. al., created a prototype of an object oriented framework to support the integration of multiple software systems used in virtual prototyping of mechanical components [13]. The goals of this work included the coupling of software systems which have different data requirements and the demonstration of the system in an industrial setting for the development of electronic spray cooling components. The framework was created using C++ and linked software systems which included a CAD system, a finite element analysis system, and a database system.

Jayaram [14],[15] designed an environment which supports the creation of device independent, customized product development tools. This architecture included design, user interface, knowledge based assistant, and virtual manufacturing environments.

Woyak, et. al. [19], describe an architecture called the Dynamic Integration System, which addresses the issues of software integration for engineering applications. This architecture is based on the concept of dynamic variables and dependency hierarchies. These provide the functionality of automated analysis execution, rapid integration, and generic interaction modules which can be reused. These features provide a system that is easily expandable and maintainable.

3. Problem Definition

With the need to further reduce the time and cost in producing a part, the use of virtual reality has been increasing. Although virtual reality techniques provide engineers with the added information needed to reduce time and cost, much virtual reality based software for design and manufacturing is limited in its expandability, customization, and usability with current CAD/CAM systems.

Three-dimensional analysis of models in a virtual environment provides the user with a more natural environment for inspecting a proposed model. This can enhance the user's ability to make decisions on appearance and functionality of a part that may not be possible on traditional computer-aided design systems. However, this only provides the user with the ability to view the model, not

change it, since the system is not tied directly to the CAD system. The user must go back to the CAD system, make changes, and then view the system in the virtual environment again. Virtual flythroughs of manufacturing simulations allow users to view a manufacturing plant in operation. Human models can be inserted and programmed to behave in the manner a real human worker is expected to behave. These manufacturing simulations may not be realistic. Ideal speeds of assembly lines, actual workspaces of workers, and other ergonomic issues may be difficult to test without an actual human in the loop performing the tasks that will be required. Much of today's virtual manufacturing is actually a manufacturing simulation of the processes involved. User interaction with the machines and parts is very limited, if provided. Interaction with manufacturing equipment is often dealt with using a mouse and keyboard approach on a two-dimensional screen, not in an immersive three-dimensional environment. Issues such as assembly constraints and ease of assembly are often not dealt with. One additional problem with many of today's virtual manufacturing software systems is the lack of customization of the environment. The user is not provided with the actual machines that are present in the factory, nor is there the ability to create them.

Although many of the problems addressed above have been researched on an individual basis, an ideal tool for engineers would be a single system that incorporates all of the virtual prototyping techniques discussed, tied directly to the CAD/CAM system being used to create the models. This would provide the engineer with a system that uses one unified model representation throughout the design cycle of the part. Most of the existing systems are not directly linked into current CAD/CAM systems, but are stand alone systems, requiring a transfer of data from the design system to the analysis system. An additional problem with many current systems is nonexpandable architecture upon which they are built. These systems do not provide for the expansion of the system by the user. Although there is research being conducted by the companies producing the various virtual reality based tools to expand the capabilities of the software, the endusers of the system do not have this freedom to expand the system. It was the objective of this research to design and implement a system called VEDAM, Virtual Environments for Design and Manufacturing, which allows designers to incorporate virtual reality techniques in the design and process planning stages of the product, while providing a flexible, expandable, and customizable environment directly linked to a parametric CAD/CAM system.

4. Proposed Solution

As stated earlier, there are several virtual reality techniques that can assist in the design and manufacturing planning of a product. These include virtual design, virtual assembly, virtual manufacturing, and human integrated design. A

valuable design system for engineers is one that will support all of these tools, yet be compatible with current parametric CAD/CAM systems. The architecture behind such a system should allow for the expansion and customization of the virtual environments to suit the engineer's needs. An overview of a proposed system, VEDAM, is shown in Figure 1. This figure shows the VEDAM system and its components, the Machine Modeling Environment (MME), the Virtual Design Environment (VDE), the Virtual Assembly Environment (VAE), and the Virtual Manufacturing Environment (VME). The MME is an environment that allows the user to create the actual machines found in the factory that will be used to produce the product. The VDE is an immersive, virtual reality based design environment that allows the user to view, scale and modify a parametric model designed in a CAD system. The VAE is an immersive, virtual environment that allows the user to analyze the assembly of parts through the direct manipulation of the parts by the user [10],[11]. The VME is an immersive virtual environment that allows the user to analyze and develop process plans using a virtual factory that replicates and functions as the real factory.

VEDAM interfaces with the parametric CAD/CAM system through the main interface. During a design session, the user has the option of entering into one of the environments via the main graphical user interface to test designs or manufacturing ideas. All required data from the CAD/CAM system then passes into the virtual environments. Upon exiting the virtual environments the user has the option of passing data back into the CAD/CAM system. VEDAM, combined with a parametric CAD/CAM system, provides a complete system for engineers to evaluate potential designs and process plans.

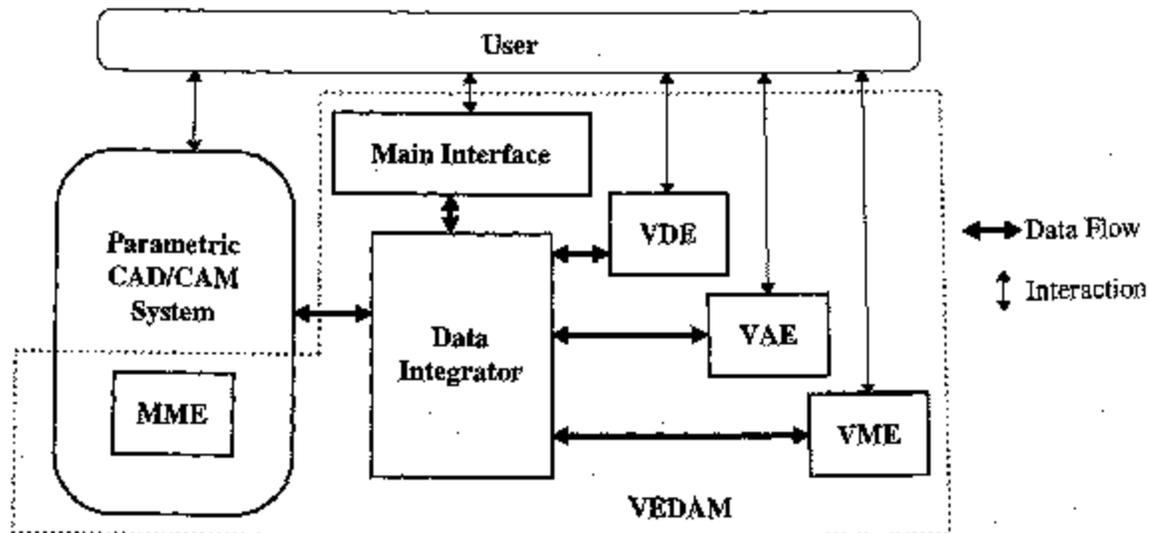


Figure 1: VEDAM System [1],[5].

5. Object Oriented Analysis

Based upon the initial concept discussed above, object oriented methods were used to create a prototype framework of the VEDAM system. Object oriented methods were used due to the complexity of the system to be designed. Traditional algorithmic decomposition of the VEDAM system would produce a very unmanageable, inflexible, and unexpandable code. By breaking the overall system down around individual classes that communicate with each other, each class can be separately created, tested, and added to the existing classes. This allows for very manageable and expandable code.

The initial object-oriented analysis of the system required a description of the functional usage of the system by a user. It could then be analyzed to create the classes needed for the VEDAM system. For this analysis, it was assumed that Pro/Engineer would be the CAD system being used by the product development team. The description used is as follows.

The first step in using the VEDAM system is the setup of the tooling capabilities of the virtual manufacturing environment, which is necessary to initialize the system. This involves the creation of the machines, tools, fixtures, etc., that are found in the actual manufacturing environment. After this is complete, the VEDAM system can be used for product development.

The user starts by creating a Pro/Engineer model of the machine. Proper hierarchy of parts and assemblies must be maintained to assure proper and accurate movement of the axial components of the machine. The user then enters the machine modeling environment. After the main components of the machine are created, the next step is the association of assemblies and parts of the model to the actual components of the machine.

First, the axial movements of the machine are associated with the corresponding assemblies of the model. Additionally, the user must specify the zero or home position of the assembly as well as the maximum distance which that component can move. Upon completing this, the user interaction capabilities with the machine are set up.

The user now adds switches, buttons or levers to the model to design the controller for the machine. These can be either user defined or selected from a library of components. If the user elects to create a button or switch, he will enter a section of Pro/Engineer specifically created for this, which provides utilities for creating all types of switches or buttons, including number of positions, colors, etc. The switches or buttons added to the machine are assembled in the same manner other parts are assembled. After adding these buttons, the user must then associate the buttons with the actions of the machine, such as emergency stop, spindle speed, or machine on/off.

The next step in setting up the virtual machine is setting machine parameters,

such as maximum and minimum feed rates, spindles speeds, and maximum power output. The user can now create the controller for the system, which is responsible for interpreting the numerical control code created by the user as well as displaying key information. The option exists for selecting different levels of numerical control to be supported. Moreover, the user has the ability to add or subtract machine specific commands, and can lay out the digital readout of the control display.

Next the user specifies the bottom and front of the machine, saves the complete setup, and creates the fixtures that can be used. This includes setting up how each fixture attaches to the machine and how it is adjusted. After the fixtures are created the type of cutters that can be used with this type of machine must be specified. This information can be coupled with that of Pro/Engineer. The user now has the option to create another machine or create the factory if all of the required machines have been created.

The user now enters the size of the factory floor and the location of walls and obstructions. Then the user selects a machine to add to the factory, positioning it and naming it for reference. Additional machines are positioned on the factory floor until the factory is complete and the setup saved. This setup is performed only once to initialize the system.

The system is now ready for use as a product development tool. The first step in using it for virtual prototyping is the creation of a parametric model using Pro/Engineer. A part or a series of parts can be created and assembled. Several options are then available depending on the goal of the user at the time system is engaged: virtual design, assembly, manufacturing, or a combination.

To use the system for virtual the VEDAM system is launched using the graphical user interface. The interface will present options for setting up the VEDAM system. The hardware that will be used, including tracking devices, gloves, and helmet mounted displays, must be set up. The various models used by the environment are then chosen, including the model for analysis and the human model. Once this is complete, the user will have the option of saving the setup for future reference, and can start the virtual design environment.

After putting on the required hardware devices, the user will be immersed in a virtual environment that contains the model being analyzed, either a part or assembly, a human model that reflects the motions of the user, and some surrounding environment models. The model can be picked up and viewed from any angle. If the model is either too large or too small to be analyzed, the model can be scaled using a three-dimensional graphical user interface. If the model is found to be unsatisfactory in its current form, the parameters and corresponding values can be viewed using the GUI. By selecting a parameter and entering a new value, the shape of the model can be modified. The value

entered is transferred back to Pro/Engineer, regenerating and transferring the model back to the virtual design environment so the user can view the modified part. This process can be repeated until a final form of the model is complete. The model is then saved again using the GUI.

If the user wants to study the assembly procedures of a series of parts, he can enter the virtual assembly environment. If the setup procedures have not already been created, those must be done at this time. Upon launching the virtual assembly environment, the user will be immersed in a virtual environment that contains the model to be analyzed, a human model representing the user, and a virtual workshop. The model will be presented in an unassembled form. The base part will be given to the user, attached to a tracking device that can be picked up by the user. The remaining parts will be laid out in a series of bins in front of the user. The user then can pick up the first part to be assembled using a normal gripping motion. When the system has determined that the part has been picked up, the user can move the part. The part, with the base part, is assembled using the constraints that the user sets up during the assembly procedures of Pro/Engineer.

When the system has determined that the user has met all of the assembly constraints sufficiently, the part snaps into place. At this point, the user can save the swept volume that was created by the motion of the assembling part as a part positioned relative to the base part. After saving the swept volume, the assembly process can be continued by picking up the next part and placing it into position. This process continues until the user exits the virtual assembly environment, elects to start over, chooses to go back a few steps, or completes the model. At any time during the assembly process the user can release the part being positioned to pick it up in another position. When complete, all of the parts that were created as swept volumes can be analyzed and have their color and transparency levels changed for analysis. This assembly of parts can also be saved for future use.

Upon using the virtual manufacturing environment, there are additional steps that must be taken prior to entering the immersive virtual environments. The user must go through the procedures in Pro/Manufacture to create all of the necessary NC codes for the parts that will be analyzed in the virtual factory, as well as the cutters and fixtures that will be used. VEDAM can now be used for manufacturing analysis. The user sets up the hardware, the models being analyzed, selects the factory to use, and the human model. The virtual manufacturing environment can then be started.

Once immersed into the virtual environment, the user is positioned inside the virtual factory with all of the stock/parts that are to be analyzed positioned on a workbench. He selects the stock, picks it up, fixes it to the appropriate machine,

and operates the virtual machine as if it were real, including turning it on, selecting a tool and putting it on the machine, loading the part program, setting the zero point, and running the program. The cutting process can be analyzed by watching the material removal as well as storing cutting properties. This includes cutting forces, cutter wear, quality of cut, and machining times. The part can then be taken off the machine and reattached to that or another machine to prepare for next process to be analyzed. This continues until the final part is completed.

If, after analyzing the cutting process, it is determined that a part needs to be modified, the user can enter the virtual design environment and make any parametric design changes needed. These changes are then transferred back to Pro/Engineer, the part is regenerated, the manufacturing code updated, and the modified part and code sent back to the virtual manufacturing environment. The process is repeated to optimize the process plans for the various parts. Once the process is complete, the model can be saved after each step of the process plans. This provides an accurate representation of the part at each stage of the process, to be used as a reference for quality control issues.

Upon analyzing the above functional description of the VEDAM system, a set of classes and class relationships are identified. These can be seen in Figures 2 through 5. Each figure shows a key group or key set of interactions of classes. Figure 2 shows those classes that are mostly related to the user of the system. Figure 3 shows those classes related to the machining operations. Figure 4 shows those classes that are related to the assembly process, and Figure 5 shows how each class that has a visual representation in the VR environment uses the geometry class. We now describe each class and its relationship to the other classes.

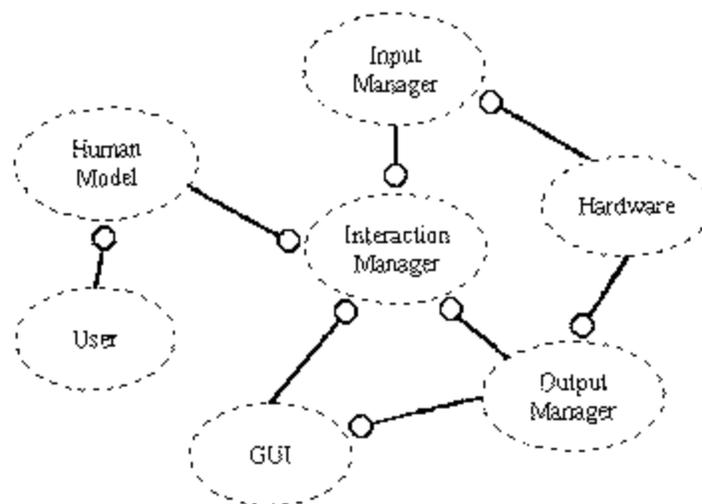


Figure 2: User Related Classes.

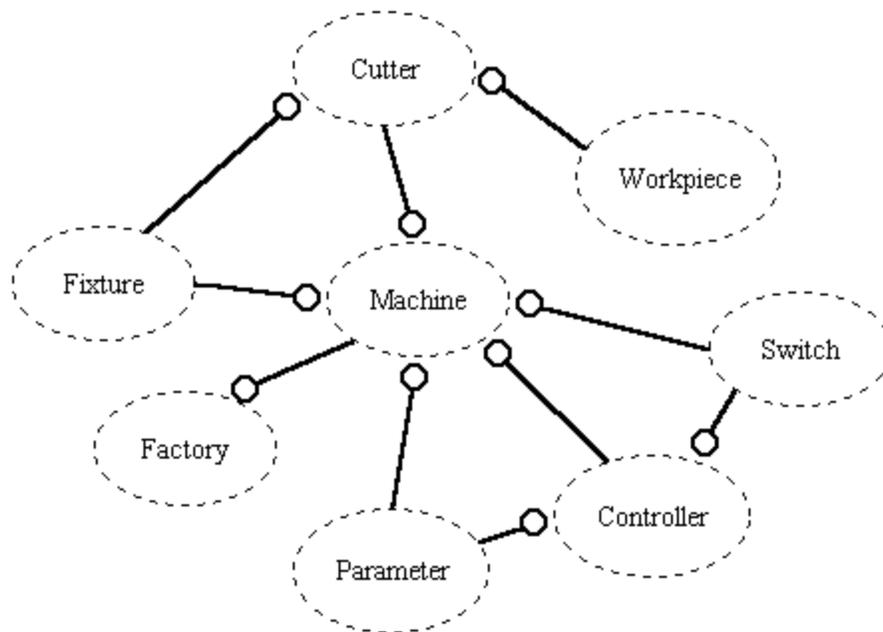


Figure 3: Machining Related Classes.

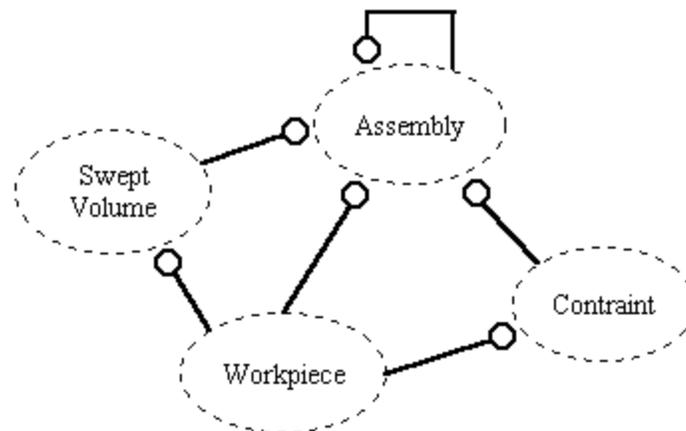


Figure 4: Assembly Related Classes.

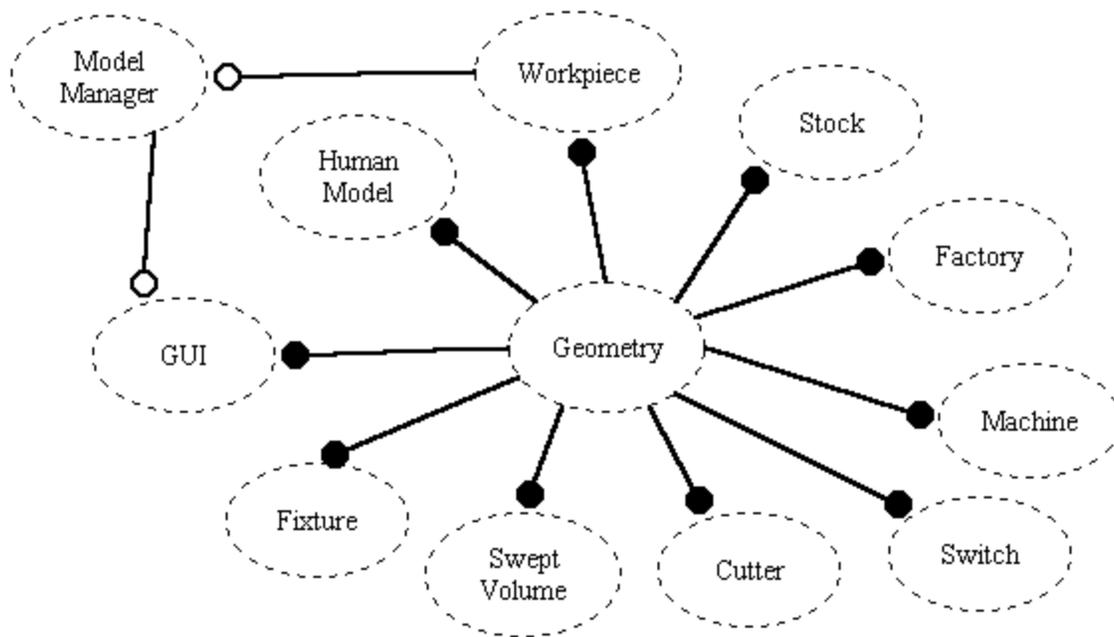


Figure 5: Geometry Related Classes.

INTERACT MANAGER CLASS

The interact manager class is the heart of the VEDAM system. This class is responsible for all interaction that occurs during the use of the VEDAM system between the user of the system and the virtual environment. As the user of the system moves within the virtual environment, the human model mimics those actions. If the user wants to pick something up or just touch an object in the environment, including the graphical user interface, the interact manager is responsible for detecting any collision and making any decisions if a response is necessary. The interact manager also contains the main loop of the VEDAM system. It must therefore communicate with a number of other classes, including the human model class, the machine class, the workpiece class, the assembly class, the fixture class, the stock class, and the graphical user interface class for interaction between the user and the environment. Based on any interaction found, the interact manager class may have to communicate with the model manager class for transferring data to and from Pro/Engineer, or the output manager for communication with hardware devices. Additionally, since the interact manager class contains the main loop of the system, it must communicate with the input manager class to obtain the latest values from the input hardware.

MODEL MANAGER CLASS

The main responsibility of the model manager class is to act as a data integrator between the virtual reality system and Pro/Engineer. When the user selects models to be analyzed, the human model to use, and the factory in which to

analyze the parts, the model manager class must take the data from the CAD models and convert it into a usable form by the VEDAM system. There are several occasions in which the model manager class must take data from the VEDAM system and transfer it in the other direction, back to the CAD system. The main occasion is when parametric design changes are made during a design session using the virtual design environment. The other occasions for data transfer are for model updates during the manufacturing process, saving versions of the model during the manufacturing process, and storage of swept volumes for later analysis. The model manager class must communicate with a number of classes during the initial integration of data between the CAD and the VEDAM systems. As the model manager reads the necessary files to create the environment for the user, the model manager class must create instances of other classes that are needed in the VEDAM system for the evaluation. Therefore the model manager must communicate with the factory class, the machine class, and all corresponding machining classes, as well as the human model class and graphical user interface class. As we shall discuss later, the model manager class also has to communicate with the geometry class since all visual objects use the geometry class.

GEOMETRY CLASS

The geometry class is used by many other classes to create a geometric model for visual representation within the virtual environments. It stores all of the needed data to create visual models, such as vertex data, colors, and scale.

HARDWARE CLASS

The hardware class is a virtual class for all input and output hardware that will be used by the VEDAM system. It is responsible for holding any information that is generic to hardware peripherals. Such information includes port numbers and baud rates. Classes that inherit the hardware class must communicate with the input and output manager classes.

INPUT MANAGER CLASS

The input manager class is responsible for getting all the information from the classes derived from the hardware class that represent an input device. The input manager receives the information from these input device classes and stores it until another class needs the information. Using this type of communication with the classes for the input devices, different input devices can be integrated without interfering with the operation of the VEDAM system. The input manager class must pass this data onto the interaction manager class so that it can be passed on to any other classes that require input device data, including the human model class for tracking the location of the user in the environment, as well as the part or assembly classes if they are "attached to" a

tracking device during the assembly process. Traditional input devices such as keyboards and mice also provide input to the system through the input manager class.

OUTPUT MANAGER CLASS

Similar to the input manager class, the output manager class is responsible for passing any information to the classes derived from the hardware class that represent an output device. The output manager receives information from another class and passes it on to the classes represented by an output device. Examples of this are a display device and a force feedback device. The interaction manager would be the main with which the output manager class communicates. As the interaction manager class detects that the human model is coming in contact with an object in the environment, it tells the output manager to pass that information on to a derived class of the hardware class. Again, using these methods the interaction between the output manager class and the interaction manager class never changes as the output hardware is changed; the protocols stay the same.

GRAPHICAL USER INTERFACE CLASS

There are two instances of the graphical user interface class. One presents the menu system of the main menu to the user at the startup of the VEDAM system. This menu allows the user to set up the environments for analysis, sets up the user and human models, and sets up the hardware that is to be used. The other responsibility of this class is the presentation of the immersive menu system to the user, which provides functions for exiting, saving parts, making parametric design changes, etc. This class has to communicate with a number of other classes, including the input and output manager classes, the model manager class, and the interact manager class. The GUI class sends messages to the input and output manager classes so that those classes know what hardware is being used by the system. The interaction manager must be notified if the user wants to exit the system or changes virtual environments. Additionally, when parametric design changes are made, the GUI class must interact with the model manager class to make the proper data integration. The GUI class uses several instances of the geometry class to create a visual menu system for the user.

USER CLASS

The user class is employed to store the information that is required by the human model, including the various measurements of the user's body, or what percentage category the user falls into relative to size. Obviously, the user class will communicate with the human model class to provide this information.

HUMAN MODEL CLASS

The human model class is responsible for providing an accurate representation of the user within the virtual environments. It contains data for representing the user as a parametric geometric model. Based on the values obtained from the user class, the parametric human model is scaled to represent the user's dimensions. This class must communicate with the model manager class to transfer the parametric human model from Pro/Engineer into the virtual environment. The human model class also must communicate with the interaction manager class. It receives the latest information from the tracking devices and other input devices to accurately mimic the user's position. The human model class also uses several instances of the geometry class to create the visual representation of the user.

FACTORY CLASS

The factory class is used to store the information about the virtual factory. This includes the number of machines in the factory, the types of machines, and the names of the machines. It also uses several instances of the geometry class for the creation of walls, floors, and other items in the factory.

MACHINE CLASS

The machine class is used to represent a machine in the actual factory. It is created during the startup of the virtual manufacturing environment, and stores all of the data that was entered by the user in the machine modeling environment needed for operation of the machine. All of the functions assigned to the particular machine in the machine modeling environment are also contained within this class. Each machine is given a unique name to distinguish it from other machines, since there can be multiple instances of any machine. The machine class uses many other classes in the process of being an operational machine. Each machine uses several instances of the switch class for representing the interactive components of the machine. It responds with some function based on interaction with switches. In addition, each machine uses a controller class for representing the controller on the actual machine. In the machine modeling environment, the user also creates fixtures that will be used by the machine. Therefore several instances of the fixture class are used by the machine class. Similarly, the machine class uses several instances of the cutter class. Since the machine is one of the main interactive components in the virtual environment, it must communicate with the interact manager class. The machine class uses instances of the geometry class for creating the geometric model.

SWITCH CLASS

The switch class is responsible for storing information concerning the functionality of a switch, button, or lever that is located on a specific machine. The type of switch, the number of possible positions, the associated parameter or function the switch represents, as well as to which machine it is assembled are all responsibilities of the switch class. When the interact manager detects that the user has touched an instance of the switch class, the switch is notified and acts according. This may involve adjusting a parameter value or turning the machine off, and also involves communicating with the parameter class or the controller class. Instances of the switch class are created by the model manager class at creation of a machine, and the geometry class is used to show the geometry of the switch.

CONTROLLER CLASS

The controller class is used by the machine class to represent the controller on the machine. The controller class mimics the controls normally found on the controller. This includes the axis control adjustments, setting the zero point of the machine as well as displaying the current position of the cutter. The controller class uses the switch class for some of this functionality. Additionally, it is responsible for reading and interpreting the numerical control code created in a CAM system, and communicating with the machine class to dictate movements and speeds.

FIXTURE CLASS

The fixture class represents a fixture used on a machine. Instances of this class are created by the model manager class at the time the machines are created. The fixture class must store how it interacts with the machine, as well as how it interacts with the parts it is intended to secure. This information is communicated to the machine class and the workpiece class. The fixture class uses the geometry class for displaying itself to the user.

CUTTER CLASS

The cutter class is used to represent a cutter that is attached to a machine. It must store all of the properties of the cutter, such as type, material, size, and the number of flutes. This information is needed when the cutting properties are calculated. The cutter class is also used by the swept volume class for determine the size and position of the swept volume. It interacts with the machine class, the workpiece class, and the fixture class to indicate when the cutter makes contact. The cutter class uses the geometry class for displaying itself to the user.

WORKPIECE CLASS

The workpiece class represents the current state of the part that is being analyzed in the virtual manufacturing environment. It communicates with the human model class when the workpiece is being handled by the user and with the fixture class when placed onto a machine, and it contains the information necessary for saving a copy of the current state of the workpiece back to Pro/Engineer. This is done through communication with the model manager class. The geometry class is used to create a geometric model of the workpiece.

STOCK CLASS

The stock class is used to represent the piece of stock from which a workpiece is being created. An instance of this class is created for each part that is being manufactured, and is also created by the model manager at the start of the virtual manufacturing environment.

ASSEMBLY CLASS

The assembly class represents an assembly of parts created in Pro/Engineer. It contains all of the necessary information needed to assemble the parts. This includes a list of all parts, the order of assembly, and the constraints required to assemble each part. The constraints are stored in a constraint class, and are used by the assembly class. The assembly class communicates with the interaction manager class for determining when parts have been picked up and are being assembled onto the current base part or assembly. It also contains the information about the swept volume created for each part assembled. This information is stored in instances of the swept volume class.

SWEPT VOLUME CLASS

The swept volume class is used to store the volume created by the sweeping motion of parts as they are assembled. As a part is assembled, data is added to this class including the position, orientation, and part. After the part has been assembled, the swept volume class creates a solid model of the swept volume and stores it as a part. This part is also an assembly member of assembly. This class communicates with the assembly class for adding itself to the assembly and to the model manager class for creating the part file in Pro/Engineer.

PARAMETER CLASS

The parameter class is used by many classes for storing parameters and their associating properties. This includes minimums, maximums, current values and methods for changing the values.

PROCESS PLAN CLASS

The process plan class is used to store the overall process plan created using

the virtual manufacturing environment and the virtual assembly environment. It contains the information about which machines are used, which fixtures and cutters are used, the order of cutting processes, the names of all part files that were created during the machining process, as well as the order of assembly. The process plan class also uses instances of the cutting properties class to store cutting times and other cutting properties.

CUTTING PROPERTIES CLASS

During the machining process, the various cutting properties are calculated and stored in instances of this class. This class would store such information as cutting time, forces, surface quality, and power consumption. All these properties are saved as functions of time. The instances of this class are used by the process plan class.

6. Initial Implementation and Results

After completing the object oriented analysis of the VEDAM system, an object oriented design produced an initial implementation of the system. The initial implementation provides the user with a tabletop milling machine, a tabletop lathe, and a water jet. These can be seen in Figures 6, 7, and 8 respectively. All three of these machines are numerically controlled through the use of word address format numerical control codes. A controller on each system provides the functionality of an actual controller. This includes buttons for power, axial movements, setting a floating zero control, loading a part program, and running the program. A graphical user interface provides the user a means for selecting NC codes. A virtual design environment, also provided by the GUI, allows the user to view parameters of the workpiece, modify the values, and regenerate the part. A menu of the VDE on the GUI can be seen in Figure 9. The user will then see the modified part in the virtual environment.

This system uses most of the classes identified during the analysis phase. The model manager we implemented currently supports the transfer of Pro/Engineer models from Pro/Engineer into the VEDAM system. These models include the machines, parts, and the human model. The interaction manager has been created to respond to keyboard entry and manages all collision detection between the human model and the environment. The input manager controls the input of the keyboard, the data flow of the Cyberglove data, and the data flow of the Flock of Birds data. The output manager is responsible for the graphical output of the system to either the monitor or the helmet mounted display. The machine class was created around the development of the table top milling machine. Generic methods were written that would provide interactive capabilities with any machine developed in the machine modeling environment. A graphical user interface class provides the three-dimensional interactive

menu system for the user.

The system was created on a Silicon Graphics Crimson workstation with Reality Engine graphics. All classes were developed using C++, and the graphics were created using Performer 2.0. The virtual reality hardware used in this implementation include a Virtual Research VR4 helmet, a Virtual Technologies 22 sensor Cyberglove, and an Ascension Flock of Birds tracking system with an ERT and six birds.

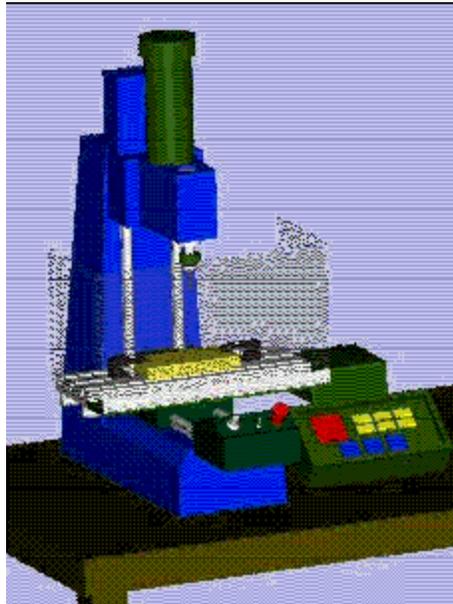


Figure 6: TableTop Milling Machine.

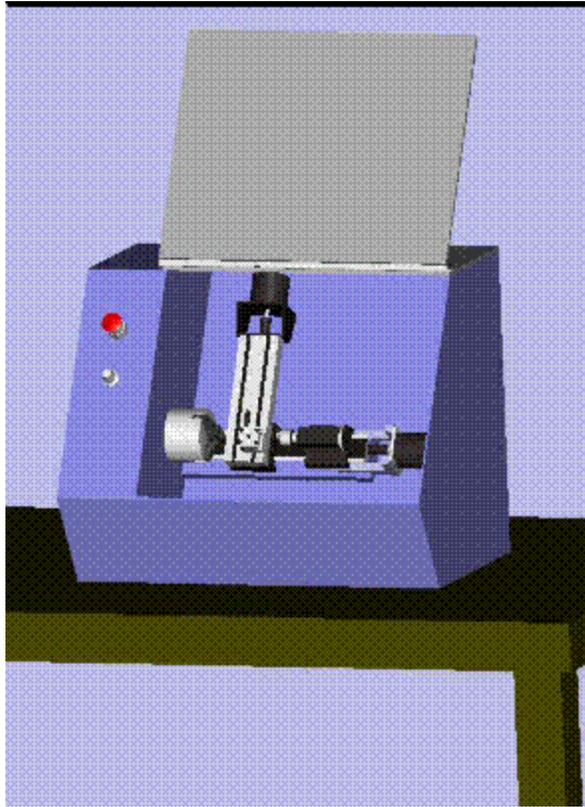


Figure 7: Tabletop Lathe.

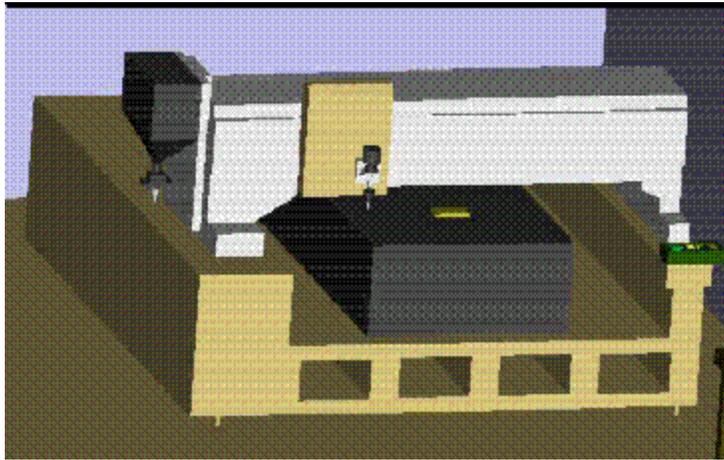


Figure 8: Water Jet.



Figure 9: GUI.

Initial tests have been run using the table top milling machine to test the VEDAM system for accuracy when compared with real machining. The tests involved the comparison of real setup times and machining times with virtual setup and machining times of a part that was created in Pro/Engineer and Pro/Manufacture. These times were also compared to those machining times provided by Pro/Manufacture and an estimated setup time that is entered into Pro/Manufacture. The same procedure and NC codes were used in both the actual and virtual environments. Details of these test are presented by Angster, et al. [5].

7. Conclusions

This paper has described the framework of a virtual reality based product development system that is capable of aiding engineers in the conceptual design and manufacturing process planning stages of a product. Through the use of object oriented methods, this framework provides an open architecture that allows the easy customization and expansion by the user. The open architecture provides the flexibility to integrate this system with any parametric CAD/CAM system using any number of input/output devices, as was demonstrated in our initial implementation. As new classes are developed, they are inserted into the system, and only minor changes to the existing classes need be made. Also, as more sophisticated methods are developed, only the class to which the method belongs needs to be modified, and without affecting the performance of the rest of the classes. These two features allow a stepwise development of the implementation.

By linking this framework to an existing parametric CAD/CAM system, engineers can immediately obtain the benefits of using virtual prototyping

techniques. The analysis of designs in a true three-dimensional environment, manufacturing the part on replications of the actual factory machines, and assembling mating parts are all valuable tools for product development. The initial implementation of this system has demonstrated the feasibility and usefulness of such systems.

REFERENCES

- [1] Angster, S.R., VEDAM: Virtual Environments for Design and Manufacturing, Ph.D. Dissertation, Washington State University, Dec. 1996.
- [2] Angster, S.R., An Object Oriented, Knowledge Based Approach to Multidisciplinary Parametric Design, M.S. Thesis, Virginia Polytechnic Institute and State University, Dec. 1993.
- [3] Angster, S.R., Gowda, S., Jayaram, S., "Feasibility Study on Virtual Reality for Ergonomic Design", Presented at the IFIP 5.0 Workshop on Virtual Prototyping, Sept. 1994.
- [4] Angster, S.R., Gowda, S., Jayaram, S., "Using VR for Design and Manufacturing Applications - A Feasibility Study", 1996 ASME Design Engineering Conferences and Computers in Engineering Conference, Aug. 1996.
- [5] Angster, S.R., Jayaram, S., "VEDAM: Virtual Environments for Design and Manufacturing", Symposium on Virtual Reality in Manufacturing Research and Education, Chicago, Oct. 1996.
- [6] Angster, S.R., Jayaram, S., "An Object-Oriented, Knowledge-Based Approach to Multidisciplinary Parametric Design", AIAA 33rd Aerospace Sciences Meeting, Reno, Nevada, Jan. 1995, AIAA 95-0242.
- [7] Barrus, J.W., "The Virtual Workshop: A Simulated Environment for Mechanical Design", Ph.D. Dissertation, Massachusetts Institute of Technology, Sept. 1993.
- [8] Bayliss, G.M., Bower, A., Taylor, R.I., and Willis, P.J., "Virtual Manufacturing", Presented at CSG 94 - Set Theoretic Modeling Techniques and Applications, Winchester, UK, April 13-14, 1994.
- [9] Bennet, G.R., "Virtual Reality Simulation Bridges the Gap Between e Engineering Database Symposium, ASME, Boston, September 1995.
- [10] Connacher, H., Jayaram, S., and Lyons, K., "Virtual Assembly Design Environment", Proc. 15th ASME International Computers in Engineering Conf.,

Boston, MA, 1995.

[11] Connacher, H., Jayaram, S., and Lyons, K., "Virtual Assembly Using Virtual Reality Techniques", accepted for publication in Computer-Aided Design, 1996.

[12] Jaques, M., Strickland, P., Oliver, T. J., "Design by Manufacturing Simulation: Concurrent Engineering Meets Virtual Reality", Mechatronics, 1995.

[13] Jayaram, U., Bakkom, J., Jayaram, S., and Seaney, K., "Integration Framework for Virtual Prototyping Software Tools", Final Report, Phase 1, Department of Commerce SBIR grant, Feb. 1996.

[14] Jayaram, S., CADMADE - An Approach Towards a Device-Independent Standard for CAD/CAM Software Development", PhD dissertation, Virginia Polytechnic Institute and State University, Apr. 1989.

[15] Jayaram, S., and Myklebust, A., "Towards a Standardized Environment for the Creation of Design and Manufacturing Software", Proc. International Conf. on Engineering Design, Dubrovnik, Yugoslavia, Aug. 1990.

[16] Narayanan, P., "An Object-Oriented Framework for the Creation of Customized Expert Systems for CAD", MS thesis, Virginia Polytechnic Inst. and State Univ., May 1993.

[17] Rosen, D. W., Bras, B., Mistree, F., and Goel, A., "Virtual Prototyping for Product Demanufacture and Service Using a Virtual Design Studio Approach", ASME Computers in Engineering Conf., 1995.

[18] Trika, S. N., Banerjee, P., and Kashyap, R. L., "Virtual Reality Interfaces for Feature-Based Computer-Aided Design Systems", accepted for publication in Computer-Aided Design, 1996.

[19] Woyak, S., Malone, B., Myklebust, A., "An Architecture for Creating Engineering Applications: The Dynamic Integration System", Proc. of the Computers in Engineering Cof. and the Engineering Database Symposium, ASME, Boston, Sept. 1995.

[20] Prosolvia Clarus AB, <http://www.deneb.com>.

[22] Resolution Technologies, <http://www.restec.com/>

[23] Technomatix Technologies, LTD., <http://www.technomatix.com/>

BIOGRAPHIES



Scott Angster

Scott Angster is currently a Research Associate at Washington State University in the School of Mechanical and Materials Engineering. He earned his B.S. and M.S. degrees at Virginia Polytechnic Institute and State University in Mechanical Engineering. His research there included the development of a software architecture that would allow the for addition of an expert system to custom computer-aided design systems. He then went on to receive his Ph.D. at Washington State University, also in Mechanical Engineering. Here his thrust of research has been the use of virtual reality techniques for computer-aided design and manufacturing applications.



Sankar Jayaram

Sankar Jayaram received his B. Tech. (1984) from the Indian Institute of Technology, Kharagpur, his M. S. (1986), and his Ph.D. (1989) from Virginia Polytechnic Institute and State University, Blacksburg. All his degrees are in Mechanical Engineering. He was the Associate Director of the ACSYNT Institute and a visiting Assistant Professor at VPI and SU until 1993 when he joined the School of Mechanical and Materials Engineering at Washington State University. He is currently an Associate Professor with research interests of VR applications in engineering, design, and manufacturing, geometric modeling, object-oriented systems for CAD/CAM, and parametric design. He directs the Virtual Reality and Computer Integrated Manufacturing (VRCIM) Laboratory at

Washington State University (www.vrcim.wsu.edu).

Contact information:

Scott Angster, Research Associate
School of Mechanical and Materials Engineering
Washington State University
Pullman, WA 99165
Email: angster@mme.wsu.edu

Sankar Jayaram, Associate Professor
School of Mechanical and Materials Engineering
Washington State University
Pullman, WA 99165
Email: jayaram@mme.wsu.edu

