

Methodology For Development Of Interactive Virtual Theory Of Machine Lab Using VrmL/X3d Capabilities

¹Dipak Vinayak Shirbhate*, ²D.S. Ingole and ³Uday A. Mehare

¹Research Scholar, ²Professor, ^{1,2}Department of Mechanical Engineering,
PRMIET&R, Amravati, Maharashtra, INDIA.

³Head of Department, Department of Mechanical Engineering,
Vikramshila Polytechnic, Darapur, Maharashtra, INDIA.

*Corresponding Author's Email ID: dipakshirbhate@gmail.com

ABSTRACT

Producing 3D interactive models is becoming a greater challenge every day. Choosing the right tool to handle the modeling process is essential if the final product is to be a VRML world, which can satisfy the user's desire for both interactivity and realism. 3DS MAX offers a potentially excellent development environment for creating high quality 3D models. This paper discuss, how the tools and techniques available within 3DS MAX can be harnessed to produce complex interactive models, which can be viewed with a VRML browser.

Keywords: VRML, Virtual Reality, interactivity, 3ds max, modeling, 3D models

I. INTRODUCTION

Section 1.1 and 1.2 set out the scope and motivation for this paper. Section 3 describes the Methodology, working platform, testing platform and the software packages used in our project. Section 2 provides a brief overview of 3DS Max modeling software. Section 4 offers an overview of VRML language. Section 5 gives a short description of what interactivity in virtual worlds means and considers the user expectations of interactivity.

What 3ds max offers for achieving interactivity in 3D models targeted for use in VRML and how we used this functionality in our project. When all objects are placed and scene is constructed, next significant step is to export the scene into VRML. Section 5 explains how this is done. Section 6 is a discussion of our experience derived from our work on the project. Section 7 is the conclusion and contains our final words on the topic.

Finally, we end this paper with future scope and references to the technical literature we used and previous work we referred.

1.1 Scope:

As the processing power of the average computer grows, in seeming fulfillment of Moore's law, so do the expectations and requirements of the average user. The demands placed on Virtual Reality programmers can be described in one word, "More": More realism, more interaction. What was excellent yesterday is unacceptable today. For practitioners who model Virtual Reality worlds, the continual demands from users mean more hours spent optimizing the code that describes these virtual environments. The modeling process is becoming more of a nightmare than an inspiring and creative challenge. So how do we shorten the modeling process and, at the same time, achieve the necessary level of interaction? One solution is to combine the use sophisticated tools, designed to support the creation of high quality models, with the techniques necessary for achieving interactivity. One such tool is 3ds max. In this paper, we investigate the full scope of the 3ds max package and we demonstrate how best the package can be exploited to achieve interactivity in 3D models targeted for use as VRML worlds.

This project work is inspired by Project Report on "Virtual Theory of Machine Lab" submitted by Mr. Amol Khurana, Mr. Anurag Goel, Mr. Gaurav Rampal, Mr. Santosh Tiwari and Mr. Hemant Ade in the year 2004 under the Guidance of Prof. D.V. Shirbhate for the partial fulfillment of award of Bachelor of Production Engineering at College of Engineering and Technology, Bahulgaon (Jh) Akola (M.S.) intended to be placed and viewed from CD-ROM media.

In this paper we will not discuss modeling techniques or techniques for code optimization. This is out of the scope of our work and moreover detailed explanations can be found in the technical literature.

1.2 Motivation

When creating VRML worlds, the first choice a programmer must make is how to do the modeling part. There are three ways:

- Visually create and edit the virtual world using specialized tools such as 3ds max;
- Create and edit VRML code with the help of a line editor (such as Notepad); or,
- A hybrid of methods 1 and 2 which means switching back and forth from visually oriented tools to line editors.

Our goal was to achieve high quality interactive 3D models using the first method - depending only on the visual environment of the 3ds max modeling software.

The objectives were:

- To minimize modeling time by using the visual environment for creating models. This approach would reduce the time spent in the coding phase of our projects. As a bonus, debugging time would be reduced to a minimum.
- To achieve necessary interaction in the created virtual environment through innovative use of the tools provided by the modeling software.
- To increase and extend the quality of models currently possible in a VRML world.

II. 3DS MAX

3ds max is an integrated environment for creating professional-quality 3D models. Photo-realistic still images and film-quality animations can be created on a PC using these models. 3ds max supports the construction of various kinds of 3D objects. Landscapes of any type can be created from these objects and characters. All of these objects can be arranged in settings and environments. These environments form the basis of the scenes. The characters and objects can be animated - setting them in motion, making them speak, sing and dance or kick and fight. Then - the whole virtual world can be captured as a film sequence.

3ds max is a rich modeling environment that supports a wide range of modeling techniques from low polygon modeling to modeling with compound objects to mesh modeling - all the way to NURBS modeling. Once our basic models have been constructed, we can assign different textures to improve the user's perception of realism. To complete the scene, different light sources (there are various kinds) can be added to illuminate the scene. Cameras can be added to capture other parts of the scene at different time intervals.

III. METHODOLOGY

The entire creation of Interactive Virtual Theory of Machine lab can be divided into following steps/stages

- **Solid Modeling**
- **Simulation**
- **Representation**

Each step explain in brief as below

Solid Modeling: The first basic step to create *Virtual World* is to create 3D models of the mechanism which needs to be displayed in the Virtual Lab. Any commercially available software like **Pro/Engineer, AutoCad** or **Catia** Etc. can be used for this purpose. We preferred, Pro/E for its parametric capabilities. The dimensions of the parts were transferred using reverse engineering techniques. The activity of solid modeling can be roughly divided into two stages:

- Part modeling
- Assembly

Following mechanisms were chosen for Virtual Lab:



Fig. 1: Withworth Quick Return Mechanism

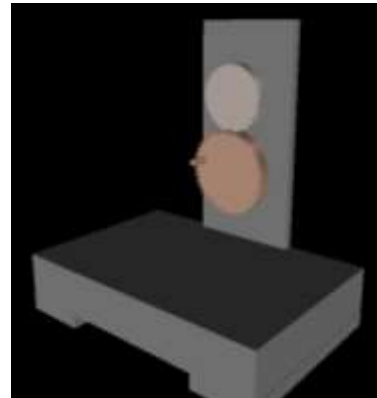


Fig. 5: Helical Gear

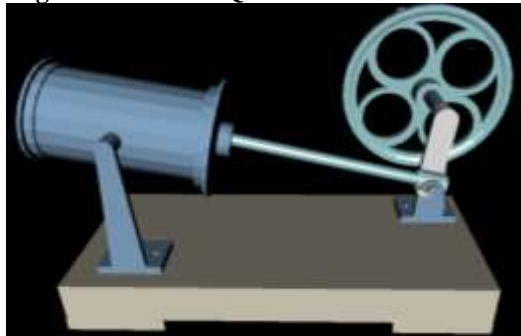


Fig. 2: Oscillating Cylinder

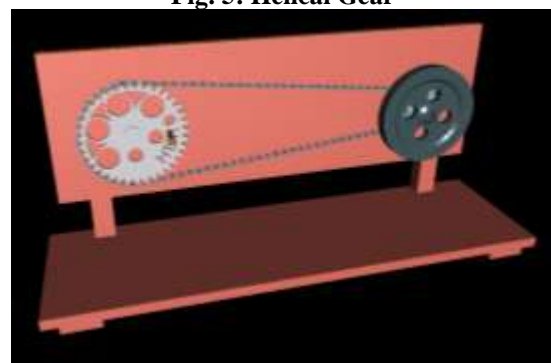


Fig. 6: Chain Drive

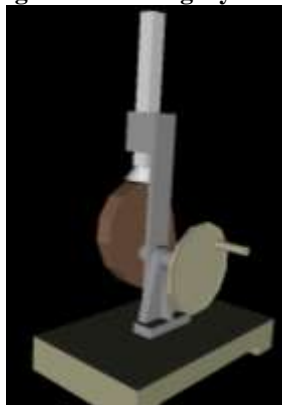


Fig. 3: Cam and Flat Base Cam Follower

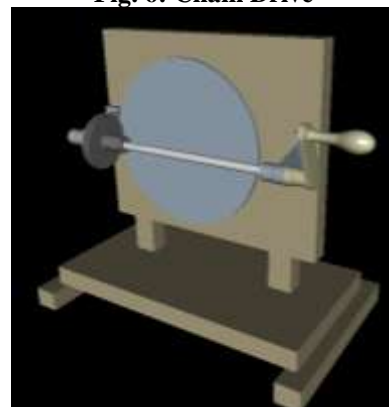


Fig. 7: Friction Wheel Drive

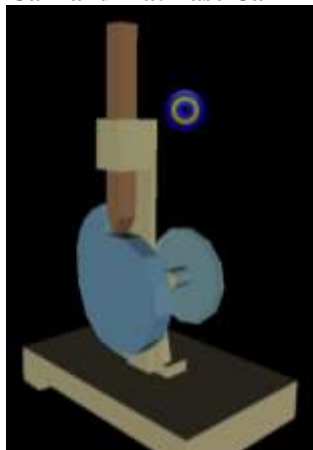


Fig. 4: Cam with Knife Edge Follower

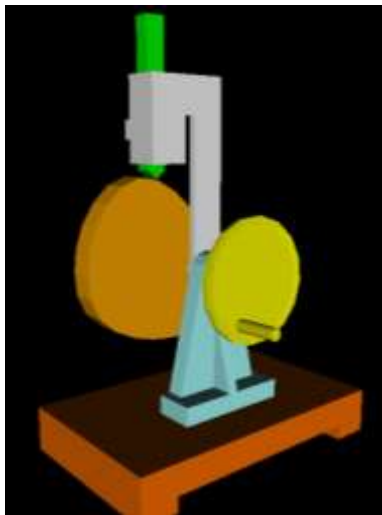


Fig. 8: Cam Offset Follower

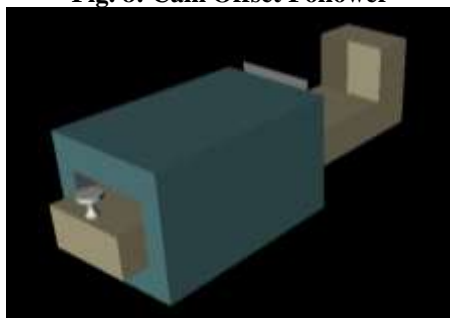


Fig. 9: Sliding Pair

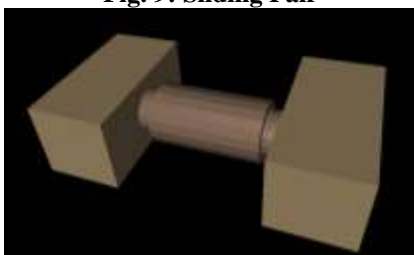


Fig. 10: Turning Pair

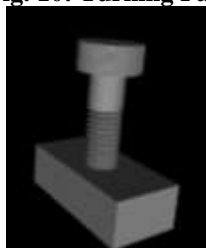


Fig. 11: Screw Pair

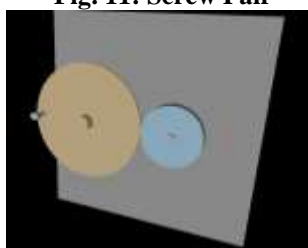


Fig. 12: Friction Disk



Fig. 13: Epicycloids Gear

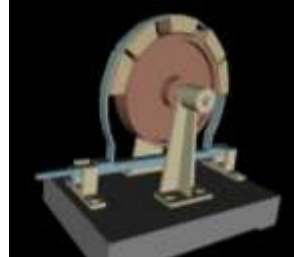


Fig. 14: Block and Band Brake



Fig. 15: Crank and Slotted Lever Mechanism

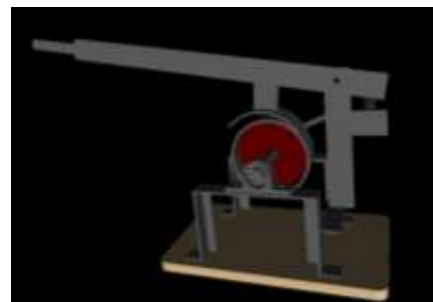


Fig. 16: Shoe Brake

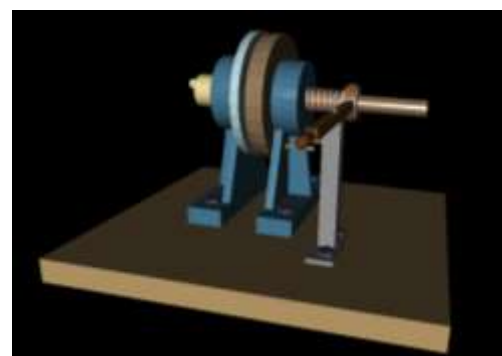


Fig. 17: Disk Brake

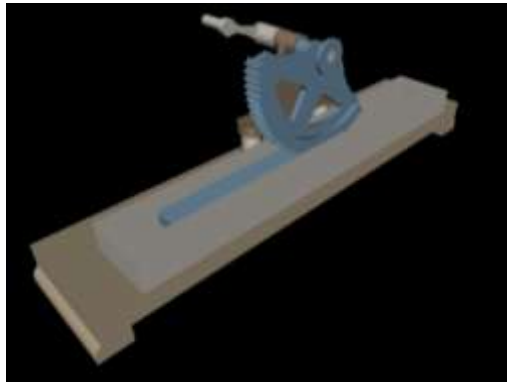


Fig. 18: Rack and Quadrant Mechanism

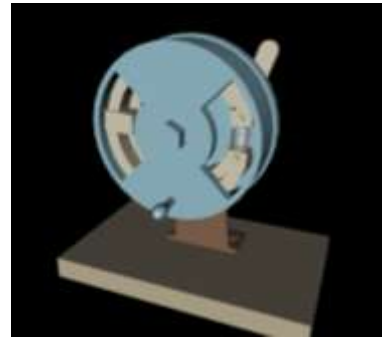


Fig. 23: Internal Expanding Brake

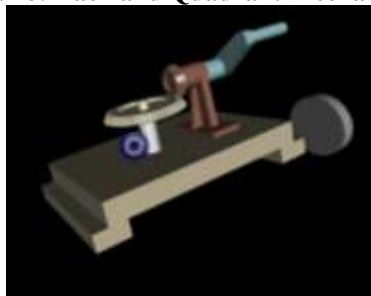


Fig. 19: Bevel Gear

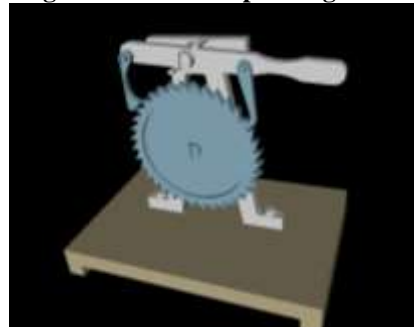


Fig. 24: Pawl and Ratchet Mechanism

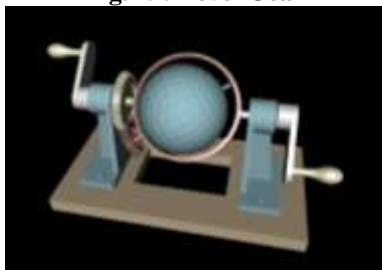


Fig. 20: Cycloidal Gear

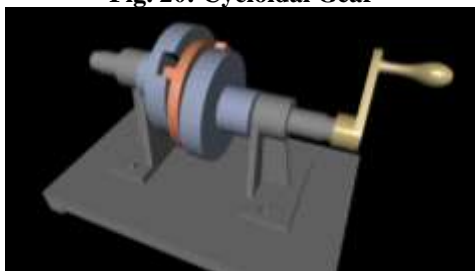


Fig. 21: Oldham Coupling

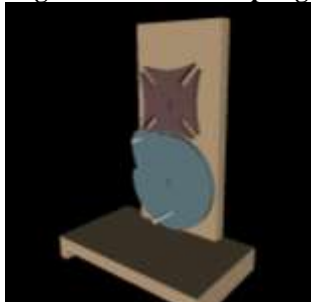


Fig. 22: Geneva Mechanism

Simulation: Simulation here means Graphical Simulation i.e. giving required motions to the links of Mechanism imported from solid Modeling package. 3DS Max Studio was used for this purpose.

Representation: Once simulation is complete, model can be exported to .wrl file format for final representation in web browser.

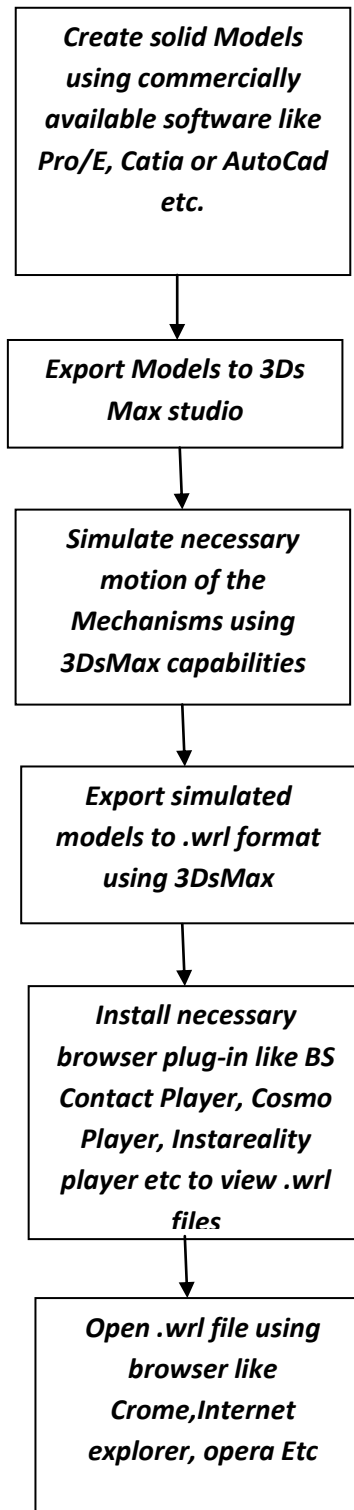


Fig. 25: Step by Step Procedure Adopted for Creation of Virtual Lab

IV. VRML

VRML stands for Virtual Reality Modeling Language. VRML is an ASCII-based open, nonproprietary language. This means that it can be used by anyone without licensing. It has been officially adopted by International Standardization Organization (ISO). The current version of this language supports animation, spatial sound, collision detection and scripting. Virtual reality is defined by a VRML file which consists of a multi-tree of nodes. Each node is of a predefined type but note that new types of nodes can be created.

The *Virtual Reality* aspect of VRML is centered on the metaphor which it pursues: human space. That space is 3-dimensional and defines the ways we move in it, perceive it and interact with it. VRML accordingly includes many of the things that are required in making the virtual world: a way of describing the geometry which creates the objects and spaces we move around in - light, texture and sound. We can approach and view the objects from different angles. We can hear the sound from different positions.

The second part of VRML, *Modeling Language*, describes the process of making a virtual world. While much of VRML is composed and optimized in text, VRML authors typically use other modeling programs for designing and creating 3D geometry in a graphical format. Later, these models, which will constitute a Virtual World, are translated to VRML and viewed using a VRML browser. The browser renders the VRML in real time and turns the code into perceivable space with which we can interact.

3ds max supports the modeling process in two ways:

- With a modeling environment capable of producing high quality 3D models
- With VRML helper objects

Helper objects are very important in VRML worlds. Without them, the user is still able to move around the virtual world but interaction is limited to looking at objects from different angles. Integration of sound, touch and other spatial perceptions can only be accomplished by using helper objects - which significantly extend interactivity.

V. INTERACTIVITY

Interactivity in Virtual Reality worlds traditionally means employment of *three* of our five senses:

- Sight
- Hearing
- Touch

Combining targeted impressions on these senses produces perceptions of space and interaction with objects located in that space. For example, incremental enlargement of an object produces a perception of movement towards that object.

What a user expects to do in a virtual world is to move freely about, manipulate objects as one does in the real world and experience a spatial sense of sound. A spatial sense of sound means that the sound has a source fixed at a single point (a node) in the virtual environment. Moving towards that source increases the volume of sound and moving away decreases the volume. Also, the perception of the source of the sound is experienced relative to the user's position in the world. By shifting position in the virtual environment, the user's perception of the direction of the sound source changes.

User interaction in VRML world is gained by using standard predefined VRML sensors. Sensors can be considered as special kind of nodes designed to react when properties of the Virtual Environment change or when a user operates a sensor in a predefined way.



Fig. 26 VRML helper Objects in 3ds max

In **3ds max**, interaction in a VRML world is achieved by adding VRML helper objects. During export, the helper objects become VRML nodes. The following helper objects are predefined in **3ds max**:

Anchor - creates a link in a VRML file. The link is embedded in an object;

Touch Sensor — (touch sensor) represents an area of sensitive space that, when touched by user, triggers event. There is a difference between Anchor and Touch Sensor.

Sound and Audio clip — these two helpers allow the insertion of spatial or ambient sounds in a scene. Sound helper is always used in combination with Audio clip, while Audio clip can be combined with Touch Sensor.

Proxy Sensor - (proximity sensor) represents an area of sensitive space that, when activated by approach of user, triggers an event;

Time Sensor — (time sensor) is used for adding time-based animation controls - such as the start and end frames for a particular object's animation;

Navinfo — (navigational information) provides navigational information of virtual space. This helper directly influences the way a user moves around the virtual world. Every scene should have this helper included;

Background - the Background helper allows the creation of a Sky and/or Ground backdrop for the virtual world. This produces very simple and plain "Earth and sky" perception.

Fog - enables specification of the colour and range of fog in a VRML world;

LOD - (Level of Detail) Lod helper allows for different levels of detail and complexity in an object depending on a user's distance from the object. When the user is closer to the object, the browser renders the object with a higher quality - producing more details. This helper can be used for optimization of performance;

Billboard - permits creation of geometry that is camera-aligned in the VRML97 browser;

InLine - lets the user reference another VRML97 file. This second file would be included in the user's world when loaded into the VRML97 browser;

VI. EXPORTING 3D SCENES FROM 3DS MAX TO VRML

Once a scene is created and everything is in place, the scene is ready to be exported to VRML. **3ds max** has a built-in VRML exporter. In version 5, **3ds max** creates VRML files (.wrl) that are compatible with the VRML97 standard. When exporting to VRML, the user has the opportunity to configure a number of important parameters in the VRML97 Exporter dialogue box shown in Figure 8.

Here is explanation of the options and their use:

- *Normals* - Creates real normals for objects, potentially making VRML files larger. This option is used for exporting 3ds max geometry, which uses smoothing modifiers. When used, it significantly improves the visual quality of objects, but, as stated, makes VRML files *larger*. This option should be used in combination with the Primitives option for the best results.
- *Indentation* - Indents the VRML source code to make it easier to read.
- *Primitives* - Exports VRML primitives instead of **3ds max** primitives. VRML predefines certain types of objects - such as sphere for
 - *Color per Vertex* - Exports vertex colors for geometry. If this option is used, the Vertex Color Source section below can be used to tell the exporter whether to use max's vertex colors or to calculate colors on export, based on lighting and the object's materials. For our projects, we did not need this option because objects were mostly covered by different materials stored as images.
 - *Coordinate Interpolators* - Enables export of animations that use coordinate interpolation, not just simple transforms. This option is mandatory with certain modifiers such as Morph. Using this option can result in very large VRML files, because the exporter has to calculate position of every vertex. Coordinate interpolation is incompatible with certain types of animation, such as animating a mesh that changes size between frames. The exporter gives warning, if this option was checked and an incompatible type of animation is detected.
 - *Export Hidden Objects* — If this option is checked, the exporter will export hidden objects which will be visible in the VRML file.

This option is sometimes confused with the Renderable option in certain objects such as Splines. Splines are by default not rendered, and so are also hidden. If Splines are used, the Renderable option

VII. DISCUSSION

Based on our work on *Interactive Virtual Theory of Machine Lab* we have drawn the conclusions, which will be presented in the following text

Creating models using visual environment of **3ds max** does shorten the time of coding but only in cases where good model design was created beforehand. A general rule of thumb is that before to any coding begins; there should be an analysis time. The same applies here. Model should be first drawn on paper and only then modeled in modeling environment. Otherwise, too much time is spent in the *search for good design*.

The interactivity achieved in our created worlds was found to be satisfactory. Animation and spatial sound were extremely easy to create and functioned in the required way.

Working with sensors that are linked together, (see above in examples of Touché Sensor and animation and Touch Sensor and Sound node) showed that the automation of creating ROUTES in VRML functions in the required ways. This is very useful when working with scenes containing many different sensor nodes.

The export utility that is integrated in **3ds max** showed satisfactory results. All primitives were exported with high-quality results. Some results were achieved by exporting extended primitives with slightly bigger file sizes. Compound objects (Boolean type and Loft type) were also exported without problems and the size of exported files was shown to be directly dependent on the objects that constructed the scene, and much less from the operation used. NURBS objects were also exported faultlessly, and exhibited high-quality result. Objects that cannot be exported are: Particle Systems and Space Warps. These objects would have been excellent for producing different weather conditions. All exported objects maintained most of their modifiers. Although modifiers increase quality of the final VRML world, they should be used sparingly, since they considerably *increase* the size of final VRML file (in some cases we observed this increase to be over 100 times!). For some applications even better export results might be needed. In this case, several third-party export utilities have been developed for specific need. These utilities come in the ready to use form of plug-ins.

The approach we have adopted produces high quality 3d interactive models. The VRML files obtained through this approach, however, tend to be *large*. If this approach is to be used for Virtual Environments viewed via such medias as the internet, then optimization methods are required. These would include controlling the number of polygons in the scenes, using simple transforms for animation, optimizing material maps and so on. Our method, however is excellent for promotional

materials that can be distributed via media such as CD-ROM and DVD.

Although it proved sufficient for the needs of our projects, 3ds max only provides a few predefined sensors. This problem could probably be circumvented by use of the Max Script - a scripting language integrated in 3ds max.

Probably the greatest advantage of this approach is that VRML worlds can be created without any great expertise in VRML. The only prerequisite is knowing how to model in the 3ds max environment. 3ds max takes care of the rest - but even so, optimization of the code is sometimes necessary and unavoidable.

VIII. Conclusion

The era of programming by writing code in a code editor is long gone. Demands on programmers are pushing them towards rapid code development. Shorter life cycle development and easier maintenance of developed systems can be achieved through different approaches ranging from computer-aided development to rapid prototyping and model development.

Demands from captivated users are driving virtual reality modelers to explore the modeling environments that can support high quality modeling and that possess integrated tool for achieving interactions in Virtual Reality worlds. Choosing a working environment is not an easy task. The chosen software tool may shorten the production period but, at worst, may considerably prolong it. If the modeling environment is too complex, the programmer will require training and practice time before becoming productive - time that we simply do not have.

3ds max is a *premium* modeling tool. Most people engaged in modeling have encountered the package at one point or another. This software is able to produce high quality and complex models and - what is more - produce them rapidly. It has a built-in VRML exporter that handles all elements of the constructed scene. As we have seen in this paper, it has built-in objects and helpers, which, if used alone or combined with one another, create the necessary perceptions of interactions. We can conclude that 3ds max presents an excellent environment for creating 3d interactive models, which can be explored in VRML.

IX. FUTURE SCOPE

VRML is a powerful tool for representation of 3D objects. Virtual reality Tool Box in MATLAB can be used to control entire world created in VRML. We are planning to develop Graphic User Interface to control VRML world through using SIMULINK of MATLAB.

REFERENCES

- [1] Project Report on “*Virtual Theory of Machine Lab*” submitted by Mr. Amol Khurana, Mr. Anurag Goel, Mr. Gaurav Rampal, Mr. Santosh Tiwari and Mr. Hemant Ade in the year 2004 under the Guidance of Prof. D.V. Shirbhate for the partial fulfillment of award of Bachelor of Production Engineering at College of Engineering and Technology, Babhulgaon (Jh) Akola (M.S.)
- [2] CD-ROM based on the project work titled “*Virtual Theory of Machine Lab*” created under the guidance of Prof. D.V. Shirbhate.
- [3] *Mastering 3ds max 4*, Cat Woods, Alexander Bichalo, Chris Murry, Sybex publishing house, 2002;
- [4] Teach Yourself VRML 2 in 21 days, Chris Marrin & Bruce Campbell, Technmedia, 1997.
- [5] Project Work Titled “Interactive Virtual Theory of Machine Lab” by Mr. Shahzaib Shaikh, Mr. Suraj Rithe and Rameshwar Ghasle under progress for Diploma In Mechanical Engineering to be awarded by MSBTE.