# Immersive Virtual Reality for Steel Structures

**Robert R. Lipman**

*National Institute of Standards and Technology*[1]
*Building and Fire Research Laboratory*
*Gaithersburg, Maryland, USA 20899-8630*
*(301) 975-3829*
*robert.lipman@nist.gov*

## ABSTRACT

This paper describes an immersive virtual reality system for steel structures. The system integrates standard models and formats for representing structural steel and three-dimensional graphic objects with a framework for device independent virtual environments. The result is an immersive virtual reality display of steel structures that includes all parts from beams and columns to bolts, welds, holes, and copes.

## KEYWORDS

CIMsteel Integration Standards, DIVERSE, steel construction, Virtual Reality, VRML

## INTRODUCTION

In recent years, immersive virtual reality (IVR) has become more commonly used in the automotive, aerospace, medical, and oil exploration industries. With the availability of more off-the-shelf hardware and software specifically designed for virtual reality, some of the barriers in implementing IVR environments, common several years ago, have been removed. Now a developer can concentrate on what's being displayed rather than how it's being displayed.

In the construction industry the adoption of 3D modeling tools has lagged other industries. The nature of a typical construction project has owners, architects, engineers, fabricators, and constructors working in a non-integrated fragmented process. This type of environment does not lend itself to IVR. However, there are some examples where IVR has been used very successfully on construction projects to solve problems related to 4D CAD (Fischer 2000).

Regardless of whether IVR will become more commonplace in the construction industry, being able to convert from standard representations of construction, such as steel, to standard 3D graphics formats is a prerequisite. This research describes a proof-of-concept where a

[1] DISCLAIMER: Any mention of commercial products is for information only; it does not imply recommendation or endorsement by NIST.

steel structure described by the CIMsteel Integration Standards (Crowley 2000) is converted to a Virtual Reality Modeling Language (VRML) file (Ames 1997, ISO/IEC 14772-1:1997) that is then displayed in an IVR system. The IVR system uses the DIVERSE framework (Kelso 2002) for extensible and reconfigurable device independent virtual environments.

## CIMSTEEL INTEGRATION STANDARDS

### Product Data Model

The CIMsteel Integration Standards (CIS/2) is the basis for the definition of the steel structures that are displayed in the immersive virtual reality system. CIS/2 is a product data model for structural steel project information. It is the result of the pan-European Eureka EU130 CIMsteel Project and has been endorsed by the American Institute of Steel Construction as the technical basis for their Electronic Data Interchange initiative. The goal of CIS/2 is to create a seamless and integrated flow of information among all project participants involved in the construction of steel framed structures. Those participants include steel designers, analysts, detailers, fabricators, and constructors.

The CIS/2 product data model deals with information about the steel structure throughout its analysis, design, detailing, and fabrication life cycle. The geometry of the structure, necessary for the virtual reality display, is only one property of the product data model. Other attributes of the structure include how parts are combined into assemblies and structures, analysis loads and reactions, material types, connection details, associations between members and drawings, and modification history.

The product data model is defined by a schema that details how all the information in the CIS/2 standard is represented and how each entity relates to each other. It also defines rules for using various entities and the information those entities contain. STEP (ISO 10303:1992), the Standard for Exchange of Product Model Data, is the technical basis used to represent CIS/2 information. STEP is a worldwide effort to develop mechanisms for the exchange and sharing of engineering data.

The format of a CIS/2 file that is imported or exported by steel related software packages is specified in a standard way to facilitate the exchange of information between various applications. The STEP Part 21 implementation method defines the physical structure of the text in a file.

Figure 1 shows a sample of a CIS/2 file for a part with a location. The file is represented in the standard STEP Part 21 format. Each CIS/2 entity instance is assigned a number indicated by a pound sign. The name of the CIS/2 entity is in upper case letters. Every entity has a number of fields that contain text strings, numeric values, boolean values, references to other entities, or null values indicated by a dollar sign. The indentation has been added to show the hierarchy and relationships between the various entities.

```
#43= LOCATED_PART(92,'92','brace',#42,#33,#20);
  #42= (COORD_SYSTEM('','Part CS',$,3)
      COORD_SYSTEM_CARTESIAN_3D(#40)COORD_SYSTEM_CHILD(#18));
    #40= AXIS2_PLACEMENT_3D('Part axes',#34,#38,#36);
      #34= CARTESIAN_POINT('Part origin',(0.,0.,0.));
      #38= DIRECTION('Part z-axis',(0.,0.,1.));
      #36= DIRECTION('Part x-axis',(1.,0.,0.));
    #18= COORD_SYSTEM_CARTESIAN_3D('','Assembly CS',$,3,#17);
      #17= AXIS2_PLACEMENT_3D('Assembly axes ',#11,#15,#13);
        #11= CARTESIAN_POINT('Assembly origin ',(720.,540.,120.));
        #15= DIRECTION('Assembly z-axis ',(-0.37139068,0.,0.92847669));
        #13= DIRECTION('Assembly x-axis ',(0.92847669,0.,0.37139068));
  #33= (PART(.UNDEFINED.,$)PART_PRISMATIC()PART_PRISMATIC_SIMPLE(#21,#26,$,$)
      STRUCTURAL_FRAME_ITEM(92,'92','brace')STRUCTURAL_FRAME_PRODUCT($)
      STRUCTURAL_FRAME_PRODUCT_WITH_MATERIAL(#27,$,$));
    #21= SECTION_PROFILE(1,'W14X158',$,$,5,.T.);
    #26= POSITIVE_LENGTH_MEASURE_WITH_UNIT
        (POSITIVE_LENGTH_MEASURE(258.48791),#3);
      #3= (CONTEXT_DEPENDENT_UNIT('INCH')LENGTH_UNIT()NAMED_UNIT(#1));
        #1= DIMENSIONAL_EXPONENTS(1.,0.,0.,0.,0.,0.,0.);
    #27= MATERIAL(1,'GRADE50',$);
  #20= LOCATED_ASSEMBLY(92,'92','brace',#18,$,#19,#10);
    #18= COORD_SYSTEM_CARTESIAN_3D('','Assembly Coordinate System',$,3,#17);
      #17= AXIS2_PLACEMENT_3D('Assembly axes ',#11,#15,#13);
        #11= CARTESIAN_POINT('Assembly origin ',(720.,540.,120.));
        #15= DIRECTION('Assembly z-axis ',(-0.37139068,0.,0.92847669));
        #13= DIRECTION('Assembly x-axis ',(0.92847669,0.,0.37139068));
    #19= ASSEMBLY_MANUFACTURING(92,'92','brace',$,$,$,$,$,$,$);
    #10= STRUCTURE(1,'cis_2','Unknown');
```

Figure 1. Located part in CIS/2 file (rearranged for clarity).

The top-level entity is a LOCATED_PART (#43) that associates a PART (#33) with a coordinate system (#42). The LOCATED_PART also refers to a LOCATED_ASSEMBLY (#20). The PART refers to a SECTION_PROFILE (#21), a LENGTH (#26), and a MATERIAL (#27). The LOCATED_ASSEMBLY also refers to a COORD_SYSTEM (#18), an ASSEMBLY (#19), and a STRUCTURE (#10).

In plain English this says that the part, which is a W14X158 wide-flange section with a given length, has a location in an assembly. The assembly has a location in the structure. According to the CIS/2 schema, all located parts must be unique. However, there can be multiple references to a single part. For a simple framed structure, there will be one located part for each beam or column, yet there might only be one referenced part if all the beams and columns have the same section profile and length. Multiple located parts can refer to the same located assembly. For example a beam and the clip angles and gusset plates at its ends can all be part of the same located assembly.

Features such as hole layouts, copes, and mitres can be applied to a part with the LOCATED_FEATURE_FOR_LOCATED_PART entity. This entity refers to the specifications and dimensions of the feature, the coordinate system to locate the feature on the part, and the located part on which they are applied. A joint system can be either a bolted or welded connection and is located on an assembly in a similar manner.

It is also important to note what is not specified in a CIS/2 file. There is no entity that says which located parts and located joint systems are associated with a located assembly. Neither is there an entity that says which features are associated with a located part. That information has to be inferred from references to located assembly in located parts or located joint systems and from references to located parts in located features.

### Data Exchange

Several vendors of steel design, analysis, detailing, and fabrication software packages have recently implemented CIS/2 import and export translators. In a typical data exchange scenario a CIS/2 file exported by a steel analysis software package would be imported into a steel detailing software package. A CIS/2 file exported by a steel analysis software package would contain information about the geometry of the structure, member section sizes, and reactions due to various loading conditions. The geometry of the structure contains only a description of the main members of the structure such as beams, columns, and braces and does not contain any information about connections or small members such as clip angles and shear plates. Once the CIS/2 file is imported into a steel detailing software package, then the connections can be designed based on the member reactions. The connections would include the details about the bolts, holes, welds, copes, and smaller connections members such as shear tabs, clip angles, gusset plates, web doublers, and others. Subsequently a CIS/2 file exported by the detailing software package could be imported into a fabrication control software package to do estimating; project, drawing, and material management; and production control and shipping.

Prior to the use of CIS/2 data exchanges the information generated by the steel analysis software package would be entered manually into the steel detailing software package or through a propriety data exchange. The manual process is prone to the introduction of errors and is not reliable. The use of proprietary formats is not open to all possible data exchange scenarios. The use of CIS/2 eliminates data exchange errors and significantly reduces the time to go from an analysis to a detailed model thus creating a cost savings that can be realized by the owner.

## VIRTUAL REALITY MODELING LANGUAGE

An intermediate step before viewing the steel structure in the immersive virtual reality display is to convert the CIS/2 representation of the structure into a 3D graphics representation. The Virtual Reality Modeling Language (VRML) is used for this representation. VRML is a non-proprietary open-standard format that is commonly used to display 3D graphics on the web. VRML can be displayed with freely available web browser plugins such as Cosmo Player and Cortona. A CIS/2 to VRML translator has been developed that is available as a Windows program or an online web service.

Typically the VRML that is generated by a CAD program or steel-related application describes the geometry of the structure through a collection of cartesian coordinates and an index array that describes how the coordinates are connected together to create faces or polygons. For those types of applications, this is the simplest way to write out geometry in a VRML file; however, it is very verbose and does not take advantage of several important features of VRML. In general the VRML exported by a CAD package has lost all of the rich information that went into creating the model and is only a representation of its surface geometry.

## Prototype Mechanism

The VRML prototype mechanism (PROTO) adds extensibility to VRML for creating new geometric nodes similar to the built-in VRML geometric primitives. This is the primary basis for developing application-specific VRML nodes to model information in a CIS/2 file. The advantage of using the prototype mechanism is that the VRML necessary to model a steel structure can be in terms related to steel such as length, width, depth, and section dimensions rather than only the coordinates that define the faces of a part. The VRML exported by CAD packages use the latter method. Previous work by the author (Lipman 2000) showed how VRML PROTOs could be used to model a complex steel structure. The same VRML PROTOs have been used to visualize steel structures on a PocketPC (Lipman 2002).

Figure 2 shows how a PROTO can be constructed to create a new VRML node called SquarePlate. The two lines after "PROTO SquarePlate" are the interface definition for the PROTO. It defines input parameters for the PROTO that are mySize and myColor. The actual VRML geometry of the object will be an IndexedFaceSet and its size is computed in the Script node. The Script node uses a JavaScript function to compute the point variable, which are the coordinates of the corners of the plate. The variable point is ROUTE'd to the Coordinate node to set the point coordinates.

```
#VRML V2.0 utf8
PROTO SquarePlate [
  field SFFloat mySize  1
  field SFColor myColor 1 0 0
] {
  Shape {
    geometry IndexedFaceSet {
      coord DEF POINT Coordinate {point []}
      coordIndex [0 1 2 3 -1]
      solid FALSE
    }
    appearance Appearance {material Material {diffuseColor IS myColor}}
  }
  DEF PLATESCRIPT Script {
    field    SFFloat mySize IS mySize
    eventOut MFVec3f point
    url ["javascript:
      function initialize() {
        point[0] = new SFVec3f (0, 0, 0);
        point[1] = new SFVec3f(mySize, 0, 0);
        point[2] = new SFVec3f(mySize, mySize, 0);
        point[3] = new SFVec3f(0, mySize, 0);
      }
    "]
   }
ROUTE PLATESCRIPT.point TO POINT.set_point
}
```

Figure 2: VRML PROTO for a square plate.

Typically a PROTO is self-contained in its own file. Figure 3 shows how the PROTO is used to model three different size plates with different colors. The third instance of SquarePlate uses the default values for mySize and myColor that are defined in the PROTO. Without using the SquarePlate PROTO, the plates could be modeled with IndexedFaceSet nodes where the coordinates of all of the corners of each plate would have to be explicitly defined and how they are connected to make faces.

The EXTERNPROTO section provides the interface definition and the URL of where the actual implementation of the PROTO can be found. By keeping the implementation of the PROTO separate from its use, the implementation can be changed without having to change any of the VRML files that use it.

```
#VRML V2.0 utf8
EXTERNPROTO SquarePlate [
  field SFFloat mySize
  field SFColor myColor
] "http://somewhere.com/SquarePlateProto.wrl"

SquarePlate {mySize 7.5  myColor 1 .7 .3}
SquarePlate {mySize 0.3  myColor 0  1  0}
SquarePlate {}
```

Figure 3: Using the VRML PROTO SquarePlate.

Similar to the method described above, the VRML prototype mechanism has been used to create application-specific nodes for geometric primitives such as parts, elements, bolts, holes, welds, and nodes. Table 1 shows several CIS/2 entities and the corresponding VRML PROTOs that have been created.

| **CIS/2 Entity** | **VRML PROTO** |
|---|---|
| PART | Part |
| JOINT_SYSTEM_MECHANICAL | Bolt |
| JOINT_SYSTEM_WELDED | Weld |
| FEATURE_VOLUME_HOLE | Hole |
| ELEMENT_CURVE_SIMPLE | Element |
| NODE | Node |
| AXIS2_PLACEMENT_3D | Axis2p3d |

Table 1: CIS/2 entities and their corresponding VRML PROTOs.

There are many CIS/2 entities that do not require a PROTO such as entities related to length, cartesian coordinates, directions, and units. The values from those entities are used as input parameters to the VRML PROTOs that are listed above. The PART entity is used to model structural members in a CIS/2 manufacturing model. The ELEMENT_CURVE_SIMPLE entity is used to model structural members in a CIS/2 analysis model. A manufacturing model contains all members and joint systems, such as bolts and welds, that would be constructed while an analysis model contains only the elements, nodes, and loads necessary to perform a structural analysis. Joint systems are not part of an analysis model.

The Part PROTO is used to represent all members of a manufacturing model regardless of whether is it a beam, column, clip angle, or gusset plate. Figure 4 shows the VRML with the interface definition for the Part PROTO and one Part. The part is a W16X36 wide flange section that is 1.5978 m long and has a notch, chamfer, and skew angle. Figure 5 shows the resulting VRML display of the Part.

```
#VRML V2.0 utf8
EXTERNPROTO Part [
  field    MFString  section
  field    SFFloat   len
  field    SFString  units
  field    MFString  feature
  field    SFColor   color
  field    SFInt32   rendopt
  eventIn  SFInt32   set_render
] "Part_p.wrl"

Part {
  section ["IB W16X36 DIM:.4028,.1774,.0109,.0075 ME CP:8 MIRROR"]
  len 1.5978
  units "METERS"
  color 0 1 0
  feature [
    "NOTCH length .0889 depth .032 TOP_EDGE START_FACE OFT"
    "CHAMFER length .0889 depth .0889 BOTTOM_EDGE START_FACE OFT"
    "SKEW angle_1 -37.38 BOTTOM_EDGE END_FACE OFT"
  ]
}
```

Figure 4: Example of using the Part PROTO.

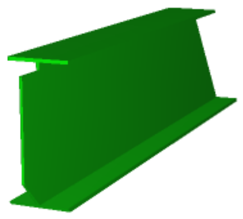

Figure 5: VRML display of a Part.

The main input parameters for the Part PROTO are its cross section, length, and features. The cross section is specified by dimensions and optional parameters such as a cardinal point, offset values, and a flag to indicate if the cross section is mirrored. The actual implementation of the Part PROTO, similar to the SquarePlate PROTO in Figure 2, contains a considerable amount of JavaScript programming. A significant portion of the code is to compute the geometry of features that can be applied to the ends of a part such as copes and mitres.

Most of the PROTOs  in Table 1 describe the geometry of an object and contain no information about their position and orientation.  Parts, elements, bolts, holes, and welds are located using the Axis2p3d PROTO, which corresponds to the AXIS2_PLACEMENT_3D entity in CIS/2.  Figure 6 shows how the Axis2p3d PROTO is used to locate a part.  The location of a part is specified by an origin, the longitudinal axis (refdir), and an up-vector (axis).  An element is located by its start and end coordinates.  The location is applied to all nodes specified by the children parameter.

```
#VRML V2.0 utf8
EXTERNPROTO Axis2p3d [
  field        SFVec3f origin
  field        SFVec3f axis
  field        SFVec3f refdir
  field        SFVec3f start
  field        SFVec3f end
  exposedField MFNode  children
] "Axis2p3d_p.wrl"

Axis2p3d {
  origin 1 0.3 -2.8
  axis 1 0 0
  refdir 0 0 -1
  children [
    Part {...}
  ]
}
```

Figure 6.  Example of the Axis2p3d PROTO.

## VRML Scene Graph

An assembly is a collection of located parts and joint systems, for example a beam, clip angles, and bolts or a column, base plate, and weld.  A LOCATED_ASSEMBLY locates that collection in the structure.  However, there is no CIS/2 entity that indicates which located parts and joint systems are contained in a located assembly.  Rather, the located parts and joint systems designate which located assembly they are part of.  Given these relationships between CIS/2 entities a VRML scene graph can be generated.  The scene graph defines the parent-child relationship between VRML nodes.

Any VRML node can have a user-defined name using DEF.  Once a node is defined then multiple instances of it can be used with the USE construct.  Using DEF and USE provides for reusing geometry and other VRML nodes, greatly reducing the size of a VRML file and speeds its processing by the VRML browser.  Reusing geometric nodes is much more efficient than explicitly creating the geometry for an object that already exists.  This method is used extensively when mapping the entities in a CIS/2 file to a VRML scene graph.

Typically in a steel structure there are multiple identical parts, features, joints, and assemblies.  In the VRML file, DEF and USE can be used to reuse identical entities.  Usually there are many identical clip angles, each with their location.  In a simple framed structure, multiple instances of an assembly of parts, features, and joint systems, could appear in several locations.  Both the clip angle and assembly could be defined with DEF and reused in other parts of the VRML scene graph. In this way, the VRML for a large structure can be developed with the maximum degree of reuse of existing components.

Figure 7 shows how a located assembly is represented in VRML using several of the PROTOs that have been developed. DEF and USE is applied to the clip angle so that it can be reused. Assuming that the size and layout of the holes that go through the clip angles and the beam are the same, then the holes can also be defined and reused. In this manner the VRML representation of an entire steel structure can be developed (Lipman 2001).

```
#VRML V2.0 utf8
# located assembly
Axis2p3d {children [
# located part - beam
  Axis2p3d {children [
    Part {}
    Axis2p3d {children DEF HOLE1 Holes {}}
  ]}
# located part - clip angle
  Axis2p3d {children [
    DEF CLIPANGLE Part {}
    Axis2p3d {children USE HOLE1}
  ]}
# located part - clip angle
  Axis2p3d {children [
    Group {children USE CLIPANGLE}
    Axis2p3d {children USE HOLE1}
  ]}
# located joint system - bolts
  Axis2p3d {children Bolt {}}
]}
```

Figure 7. VRML representation of a beam with clip angles using DEF and USE.

## IMMERSIVE VIRTUAL REALITY SYSTEM

### DIVERSE

The immersive virtual reality system at NIST consists of a two-walled CAVE or corner. Two 2.44m by 2.44m rear stereoscopic projection screens are arranged at right angles. The system is driven by a 16 processor SGI Onyx3 visual supercomputer with 4 InfiniteReality4 graphics pipes and 16 Gbytes of memory (Daunkantas 2001). CrystalEyes stereoscopic glasses with head tracking and a tracked wand are used for navigation. DIVERSE (Device Independent Virtual Environments – Reconfigurable, Scalable, Extensible) is the software system that drives the display in the CAVE.

DIVERSE is a modular group of software packages designed to make possible the creation of device independent virtual environments (Kelso 2002, DIVERSE 2003). The DIVERSE software contains both a C++ API and end-user programs to allow scientific data to be brought into an IVR system.

## Displaying Steel Structures

To display a steel structure in the form of a CIS/2 file, the file is first converted to a VRML file. There is no VRML file loader for DIVERSE so the VRML file must then be converted to Performer format. This is done with a loader from OpenWorlds that understands all of the VRML PROTOs that were developed for this visualization. Once converted to Performer format it can be displayed in the CAVE. Figure 8 shows part of an exterior steel structure that includes a staircase and connection bolts. Parts are colored depending on their cross section. Bolts are colored orange. The red sphere and cylinder extending from it, in the lower left, are a virtual wand that can point at parts of the structure and highlight them. There is a faint "ghost" image of the structure caused by some jitter in the display.

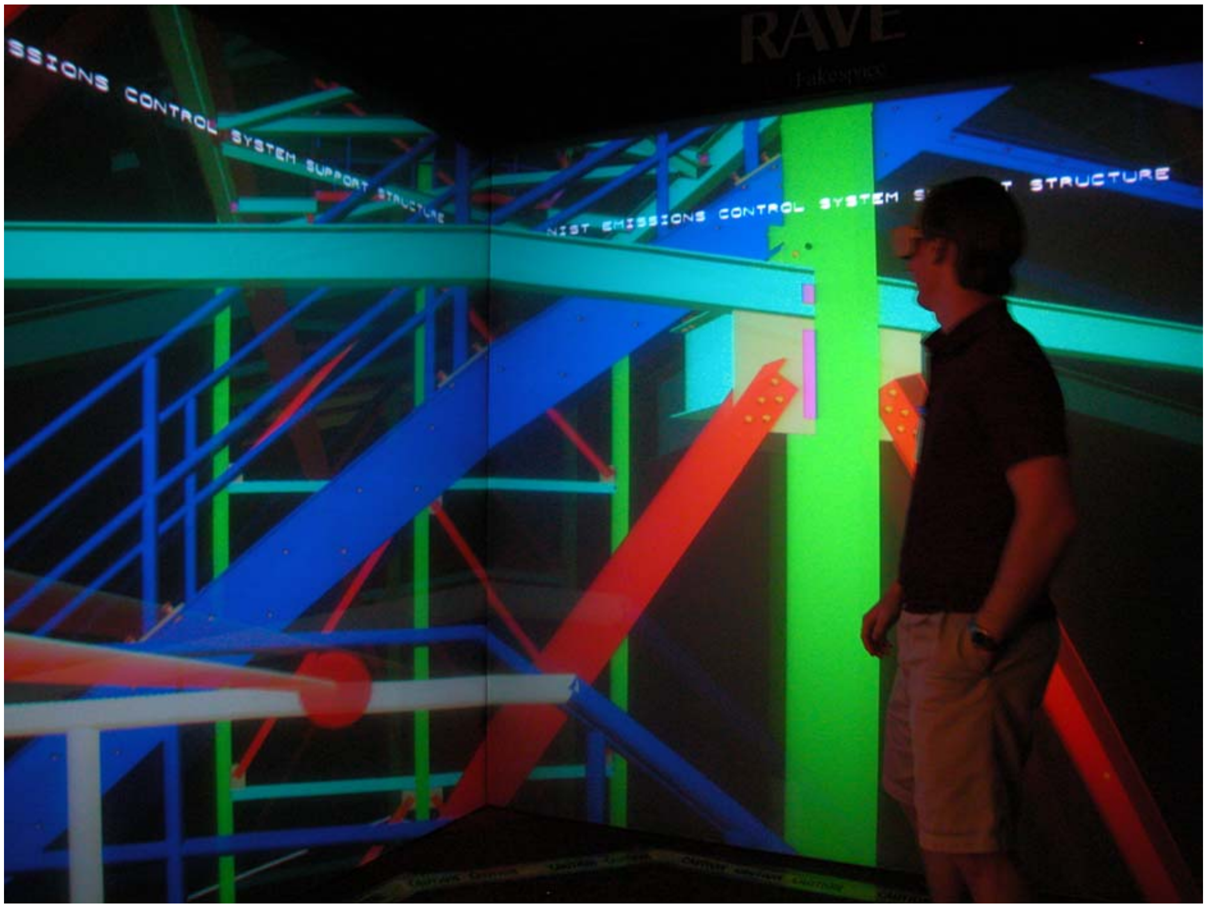
RAVE
SSIONS CONTROL SYSTEM SUPPORT STRUCTURE
NIST EMISSIONS CONTROL SYSTEM S
T STRUCTURE

Figure 8. Steel structure displayed in the NIST CAVE.

As with any IVR system, less is more. The fewer polygons that are displayed, the better the interactive frame rate will be. Higher frame rates are always desirable in an immersive environment. Several techniques were used to reduce the number of polygons in the VRML model. For very large models, details such as bolts, holes, and welds are not displayed. Typically there are more polygons in the bolts, holes, and welds associated with one beam or column than there are in the beam or column. If those details are included, then a level-of-detail node is used so that they are not visible beyond a specified distance. Therefore details that are far away are not visible.

Another detail that can be eliminated is the thickness of webs and flanges for beams and columns. At a distance, the thickness of webs and flanges is not apparent. To ignore the thickness of webs and flanges, only one polygon is used to represent a web or flange rather than many polygons to model all sides of a web or flange. This greatly reduces the number of polygons to be displayed.

A level-of-detail node can also be used for beams and columns. Up close, the thickness of the webs and flanges would be visible while at a distance the simpler polygonal representation would be used. While this is a viable alternative to reduce the number of polygons being displayed, some potentially annoying affects are seen. The distance at which the level-of-detail switches between two representations is associated with one end of a beam or column. It is possible to be very close to a beam or column when the thickness of the webs and flanges should be visible, but the lower resolution version is visible because the end of the beam or column associated with the level-of-detail is "far" away. There are also lighting issues in that the lower resolution webs or flanges use only a one-polygon representation, thus having only one normal to be used for lighting calculations. The affect is that the lower resolution webs and flanges have different shading than those were the thickness of the webs and flanges are visible. Finally one would assume that moving closer to a beam or column would switch to the higher level-of-detail. In the CAVE, there are two ways to move closer to an object. The wand is used to move large distances and small distances can be walked through within the confines of the head-tracking system. However, due to a software problem, physically walking towards the steel structure switches to the lower level-of-detail rather than the higher level-of-detail.

## FUTURE WORK

This research has shown how a steel structure can be displayed in an IVR system. The steel structure is in the form of a CIS/2 file and translated to VRML for which PROTOs have been written to facilitate the mapping between CIS/2 and VRML. The VRML file is then displayed in a CAVE that is controlled by the DIVERSE framework for virtual environments.

Currently only a static model of the steel structure is being displayed. To be of more utility, access to other information associated with the steel structure is desirable. For example a user might point to a beam or column with a virtual wand and popup some text information about that individual part or it's parent assembly. Such information might include dimensions, material grade, cross section designator, piecemarks, barcodes, CAD drawings, quantities of bolts, and loads and reactions. All of this information is already in the CIS/2 file. The current version of the CIS/2 to VRML translator can also output an XML file with all of the non-geometric information. On a PC, in combination with a Java applet the XML information is parsed and can be accessed through the VRML model to generate text reports about the CIS/2 model. With the extensible capabilities of DIVERSE, it should be possible to access the same XML file and generate a text report that can be displayed in the CAVE.

## REFERENCES

Ames A, Nadeau D, Moreland J (1997). VRML 2.0 Sourcebook (2nd edition), John Wiley & Sons, Inc.

Crowley A, Watson A (2000). CIMsteel Integration Standards Release 2, The Steel Construction Institute.

Daunkantas, P. (27 Aug 2001). "NIST Team Sees in Stereo." Government Computer News.

DIVERSE (2003) [online] http;//diverse.sourceforge.net/ [Visited 7 May 2003].

Fischer, M (2000). "Construction Planning and Management using 3D & 4D CAD models." Construction IT 2000, Sydney, Australia, April 10-11.

ISO 10303:1992 Industrial automation systems and integration – Product data representation and exchange *(*http://www.iso.ch/*)*

ISO/IEC 14772-1:1997 Information technology -- Computer graphics and image processing -- The Virtual Reality Modeling Language (VRML) -- Part 1: Functional specification and UTF-8 encoding (http://www.web3d.org/)

Kelso, John et al. (2002) DIVERSE: A Framework for Building Extensible and Reconfigurable Device-Independent Virtual Environments and Distributed Asynchronous Simulations, Proceedings of the IEEE Virtual Reality 2002 Conference, 24-28 Mar 2002, Orlando, FL.

Lipman R, Reed K (2000). Using VRML in Construction Industry Applications, Proceedings of the Web3D-VRML 2000 Symposium, 21-24 Feb 2000, Monterey, CA.

Lipman R (2001). CIS/2 and VRML Research at NIST [online] http://cic.nist.gov/vrml/cis2.html [Visited 7 May 2003].

Lipman R (2002). Mobile 3D Visualization for Construction, Proceedings of the 19th International Symposium on Automation and Robotics in Construction, 23-25 September 2002, Gaithersburg, MD.