EMULATION OF AUTOMATED STRUCTURAL STEELWORK ERECTION USING CIMSTEEL INTEGRATION STANDARDS

Vineet R. Kamat[1] and Robert R. Lipman[2]

ABSTRACT

Automation is being increasingly explored as a possible solution for safely increasing productivity in structural steelwork erection. A piece of automation equipment such as a robotic crane has no intrinsic knowledge of the steel erection process it automates. Thus, geometric and spatial information about a steel member and the motion sequences that must be executed to move that component from a staging area to its installed final location must both be programmed into the equipment. The equipment must minimally know where a steel member in question is currently staged, and what the final installed position and orientation of the member is in the erected structure. The presented research investigates the extent to which the CIMsteel Integration Standards (CIS/2) can specify product descriptions capable of supporting automated erection of structural steelwork. Algorithms to interpret steel member geometry and spatial configuration from CIS/2 files were designed. Then, a kinematically smart crane capable of accepting robot-like instructions was implemented in 3D virtual reality. The crane was programmed to utilize the algorithms to automatically extract member information from CIS/2, and to use that information to compile assembly instructions for erecting the structure in the virtual world. Based on the emulation results, it was found that CIS/2 does encapsulate the basic geometry and pose of steel members in a format that, after geo-referencing, can be used to support automated steelwork erection. However, several processing steps are necessary before the extracted data can be readily used to program automation equipment.

KEY WORDS

Automation, CIS/2, Emulation, Interoperability, Product Data Model, Structural Steelwork.

INTRODUCTION

Automation of structural steelwork erection is being actively explored as a possible solution to safely increase productivity in the steel construction industry (Lytle et al. 2004). A piece of automation equipment such as a robotic crane has no intrinsic knowledge of the process (e.g. steel erection) it automates. Thus, geometric and spatial information about a component

[1] Assistant Professor, Dept. of Civil and Environmental Engineering, University of Michigan, 2350 Hayward Street, Ann Arbor, MI 48109-2125, USA, Phone 734/764-4325, FAX 734/764-4292, vkamat@umich.edu

[2] Computer Engineer, Computer-Integrated Building Processes Group, National Institute of Standards and Technology (NIST), 100 Bureau Drive, Gaithersburg, MD 20899-8630, USA, Phone 301/975-3829, FAX 301/975-5433, robert.lipman@nist.gov

(e.g. steel member), and the motion sequences that must be executed to move that component from a staging area to its installed final location must both be programmed into the equipment. The equipment must minimally know where a steel member in question is currently staged, and what the final installed position and orientation (pose) of the member is in the erected structure. Based on these two pieces of information, automation equipment can use inverse kinematics (IK) algorithms to first move its grippers to the current location of the member, and then transport it to the pose where it is to be installed.

In automated steel construction, the position and orientation of steel members in a temporary staging area is project and site dependent, and thus cannot be automatically determined beforehand. User intervention is required to interpret and communicate such information to automation equipment. The final in-place spatial configuration (position and orientation) of a steel member, however, can be conceptually extracted automatically from a three-dimensional product model of the structure being erected (Reed 2002). The presented research evaluates this hypothesis and investigates the extent to which the CIMsteel Integration Standards (CIS/2) can specify product descriptions capable of supporting automated erection of structural steelwork.

STRUCTURE OF THE CIS/2 PRODUCT DATA MODEL

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 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.

A CIS/2 file can support three primary models of structural steel information - the analysis model, the design model, and the manufacturing model. Information on any of these models can coexist in the same CIS/2 file. An analysis model of a steel structure consists of nodes and elements and supports several static and dynamic analysis methods. A design model represents a steel structure as a design assembly to allow member and connection design. A design assembly can be partitioned into other simpler design assemblies and eventually into design parts and design joint systems. The design parts and joint systems respectively form the conceptual representations of a basic piece of steel and joint system.

A manufacturing model in CIS/2 represents a steel structure as manufacturing assemblies for the purpose of detailing, production planning, and manufacturing. In a manufacturing model, located assemblies are comprised of located parts and located joint systems that respectively represent a basic physical piece of steel and a basic physical joint system. All located items can be combined into larger located assemblies that eventually define a complete structure.

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 then appears 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 a CIS/2 File

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). These statements describe that the part, which is a W14X158 wide-flange section with a given length, has a location in an assembly that in turn is located 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, each beam or column is a located part. However, there need be only one referenced part for members that have the same section profile and length. Multiple located parts can refer to the same located assembly.

TECHNICAL APPROACH

By knowing where a steel member to be erected is currently located, and what its final installed position and orientation in the frame is going to be, a piece of automation equipment can use inverse kinematics algorithms to first move its grippers to the current location of the member, and then transport it to the location in the frame where it is to be installed. This concept of autonomous pick and place of a steel beam has been demonstrated using the NIST RoboCrane and a simple two-column frame with ATLSS connectors, though the necessary part transformations were not derived from CIS/2 data. (Saidi et al., 2005). Typically, additional spatial information such as the geometry of the partially completed structure and other existing obstructions is also required to be input to the equipment so that interference detection algorithms can work with the IK algorithms to compute a collision free path for the equipment and the member.

The final position and orientation of each steel member in the completed structure can be conceptually deciphered from a rich geometric product model of the structural frame. CIS/2, for instance, can allow the definition of all steel parts, prefabricated assemblies, and connecting joint systems that are to be erected in a steel structure. In terms of information needs, virtual pieces of construction equipment are very similar to industrial robots. The two may differ in context (real vs. virtual) and shape, but in both cases, engineers are basically interested in trying to manipulate a multiply articulated structure (i.e. a kinematic chain) and move a particular component (e.g. steel member) from one pose to another. The design and implementation of geometric information extraction algorithms and their evaluation in a 3D virtual world thus presented an effective way of validating the applicability of CIS/2 for supporting automated erection of structural steelwork.

A two step approach was adopted to achieve these objectives. In the first step, the NIST CIS/2 to VRML translator (Lipman and Reed 2003) was used to convert CIS/2 files to their corresponding VRML (Virtual Reality Modeling Language) representation. Then in the second step, algorithms were designed to extract member geometry and pose information from the VRML files. The intermediate conversion of CIS/2 files to VRML was adopted for the following reasons:

- Although CIS/2 supports explicit description of geometry, most CIS/2 parts are expressed using implicit descriptions of geometry based on a transverse section profile and a longitudinal length through which the profile is swept (Reed 2002). The VRML translator converts all implicit geometry to regular geometry facilitating member shape extraction.
- Implementation of a VRML file parser that can traverse the described scene graph to extract member geometry and pose information was relatively straightforward compared to the implementation of a full-fledged CIS/2 file parser.
- The VRML translator provides a visual interface to the underlying CIS/2 data thereby providing a means to visualize the represented steel structure in 3D.

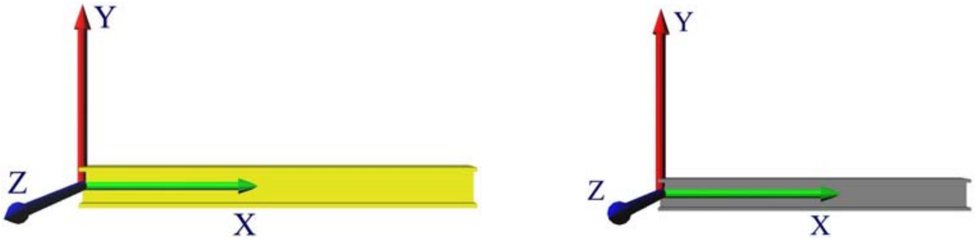
EXTRACTION OF STEEL SHAPE GEOMETRY

In a CIS/2 manufacturing model, an assembly is a collection of located parts and joint systems. For instance, a column, base plate, and welded connection could be located in the structure as a collection using the LOCATED_ASSEMBLY entity. There are no CIS/2 entities that can indicate which located parts and joint systems are contained in a located assembly. Instead, the located parts and joint systems indicate the located assembly they are part of. Using these relationships between defined CIS/2 entities, the CIS/2 to VRML translator generates a scene graph defining the parent-child relationship using corresponding VRML nodes. Any VRML node can have a user-defined name using the DEF construct. Once a node is defined, multiple instances of it can be used in the scene graph with the USE construct. Using DEF and USE, geometry and other nodes can be reused, greatly reducing the size of a VRML file and the time required to process it inside a browser for display. Since reusing geometry nodes is more efficient than explicitly creating the geometry for an object that already exists, this method is extensively used in mapping CIS/2 entities to a VRML scene graph (Lipman 2002).

While this approach is favorable from the perspective of file size and browser processing time, it is not very amenable to individual member geometry extraction required for supporting automation. When defined unit length geometry nodes are reused to instantiate specific instances of steel members, the geometry of the members is effectively being defined implicitly in terms of the stack of scaling transformation nodes that appear between the member and the referenced geometry node. This presents an identical problem to that encountered when attempting to deduce three-dimensional geometry coordinates of a member from the implicit description of the geometry in a CIS/2 file defined using a section profile and swept longitudinal length.

In order to address this issue and represent member geometry in a way that facilitates the extraction of individual three-dimensional member coordinates, an option was added to the CIS/2 to VRML translator which, when selected, collapses the scaling transformation nodes that would otherwise appear above a member’s geometry. The shape of each member is entirely encapsulated in individual geometry nodes using VRML IndexedFaceSets. Since the geometry of each steel member is represented separately even if two members are identical (i.e. same section and length), the size of the resulting file is relatively larger compared to the former scenario. However, this does not present any particular constraints for the purposes of this study.

The geometry of each member is described relative to a local origin. Figure 2 graphically presents the geometry descriptions for an arbitrary steel column and beam. In both cases, the profile of the section is coplanar with the YZ plane and the length is represented along the positive X axis. In the case of columns, the local origin corresponds to the lower end of the column when located in the overall structure. In the case of beams, the local origin in all tested files corresponds to the beam end closer to the global origin when located in the overall frame. However, this is only a convention and not a requirement of CIS/2.



(a) Column (b) Beam

Figure 2: VRML Geometry in a Local Coordinate System for Arbitrary Column and Beam

EXTRACTION OF STEEL SHAPE POSITION AND ORIENTATION

Similar to geometry, CIS/2 defines a hierarchical system of steel member locations (positions and orientations) such that an entity is located with respect to another parent-level entity to which it belongs. For example, a part may be located relative to an assembly that in turn could be located with respect to another higher-level assembly. The hierarchical description can continue until the location of all entities in the complete structure is defined relative to a chosen global coordinate system.

In CIS/2 manufacturing models, each three-dimensional location is typically defined using the COORD_SYSTEM_CARTESIAN_3D entity whose attributes are a three-dimensional point defining the origin, and a set of vectors defining the three-dimensional orientation of the local coordinate system. The final position and orientation of each member in the overall structure is thus a function of the member's geometry in the leaf coordinate system and the combined effect of all parent coordinate systems higher up in the hierarchy.

In the case of a converted VRML file, this same effect is achieved using a stack of geometric transformation nodes that each corresponds to a specific local coordinate system. The geometry of a member is located with respect to a leaf transformation node that in turn has another transformation node as its parent and so on until the top level transformation node represents the global coordinate system in which the represented structure is located.

The position of a part in the overall structure defined in a VRML file can thus be determined by computing the combined effect of all translation fields that appear in the hierarchy of transformation nodes above the definition of the member's geometry node. The important point to note is that the combined effect of all translation fields appearing above a member's geometry only places the geometry's local origin (Figure 2) at the computed location in global space. It does not necessarily orient the member correctly in global space.

In order to determine a part's correct global orientation, the combined effect of all rotation fields in the transformation hierarchy is computed. The vectors defining rotations in a CIS/2 file are represented in a converted VRML file using the axis-angle representation. In this format, a rotation is described by an arbitrary axis in three-dimensional space and the amount by which the local coordinate system is to be rotated about that defined axis. The axis itself is defined as a vector constructed from the local origin to the three-dimensional coordinate specified in the description. The CIS/2 to VRML translator was modified to collapse as many hierarchical geometric transformations (positions and orientations) as possible during the conversion process to facilitate the parsing of the generated VRML file. The structure of the resulting VRML scene graph is graphically presented in Figure 3. In the case of a CIS/2 manufacturing model, for instance, all geometric transformations were collapsed down to the individual assembly level, i.e. each primitive CIS/2 assembly was directly positioned and oriented in the VRML file's global space. This facilitated the parsing and interpretation of member position and orientation.

MODIFYING EXTRACTED CIS/2 DATA TO SUPPORT AUTOMATION

As described above, the parsing and interpretation of a VRML file converted from a CIS/2 model provides the basic information on member geometry and pose that is required to support automation. However, the extracted information must be pre-processed in the following steps before it can be input in instructions to automation equipment, and in this case, to a virtual crane:

- Reconciliation of Steel Member and Automation Equipment Coordinate Systems
- Coincidence of Member Origin with Automation Equipment Gripper
- Conversion of Axis-Angle Member Rotation to Euler Angles

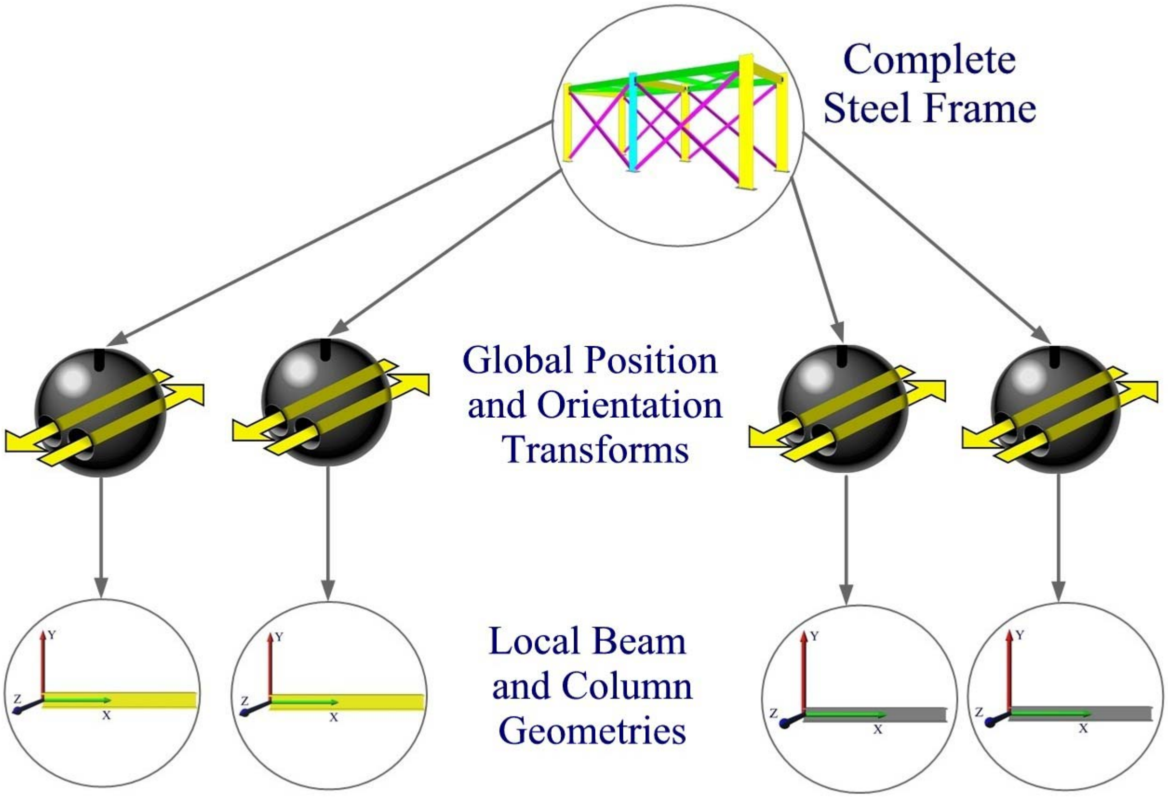

Complete Steel Frame
Global Position and Orientation Transforms
Local Beam and Column Geometries
Y
Z
X

Figure 3: Local VRML Geometry Nodes with Collapsed Transformations

The parsing and interpretation of CIS/2 information contained in a converted VRML file, and its subsequent modification thus yields the following information for each steel member contained in the represented structural steel frame:

- The name of the member
- The geometry of the member in local space
- The position of the member in global space
- The orientation of the member relative to each coordinate axis in global space

Together, this automatically extracted data comprises a significant portion of the information required by a robotic piece of equipment to automatically erect a steel member by moving it from a staging area to its installed location via a computed collision-free path.

EMULATED ERECTION OF STEEL COLUMNS AND BEAMS

The articulated crane implemented in the KineMach add-on for the VITASCOPE visualization system was used to validate the efficacy of the proposed approach. KineMach (Kamat and Martinez 2005) implements “smart”, generic pieces of virtual construction equipment and provides simple parametric text statements that can be used to issue task-level instructions to that equipment to visually depict the performance of construction work. Currently implemented generic pieces of equipment include a tower crane, a crawler mounted lattice boom crane, a crawler mounted backhoe, and a highway dump truck. Many KineMach statements are designed using standard, commonly used terminology. For instance, in the case of the crawler crane, most statements have a direct correspondence with standard crane hand signals used in real crane operations. KineMach deciphers a communicated instruction and uses IK techniques to compute the elemental motions involved in performing a construction task. KineMach then applies the computed elemental motions to the equipment’s components (e.g. boom, cabin, etc.) to depict the performance of the requested task in a virtual world. For instance, each time a steel shape is to be placed, the following statement can be communicated to KineMach requesting the instantiated virtual crane (Crane1) to perform the assigned task.

```
Crane1.PutThatThere LA2550 6 (0,6,0) (8,9,2) 45 90 4 60;
```

The PutThatThere statement instructs an instantiated crawler-mounted crane (Crane1) to perform a lift inside the virtual world. In particular, the statement instructs a crane to pick up the specified object (LA2550) and install it at the indicated target location (8,9,2) in the given amount of time (60). The statement also requires that the height of the object being lifted (6) be provided for clearance computations, and the hook attachment point (0,6,0) of the object be specified for identification of the grip locations. In addition, the object’s target horizontal (45) and vertical (90) rotations (i.e. orientations) and the minimum clearance to maintain (4) during the animated operation must also be indicated in the statement’s arguments. The efficacy of the proposed information extraction approach was evaluated by obtaining the argument values for the virtual crane’s PutThatThere statements from the geometry and pose information of steel members interpreted from a CIS/2 VRML file. The mapping of CIS/2 information to KineMach argument values is graphically presented in Figure 4.

Based on the arguments provided to the PutThatThere statement, KineMach computes a collision free path for a member from its staging area to the location where it is to be installed. The vertical clearance argument (4 units in Figure 4) is used in the path computation. For example, KineMach will raise the hook of the crane so that the member being hoisted is at least “vertical clearance” units higher than the other CAD objects it will pass over in the virtual environment. Figure 5 presents animation snapshots displaying the erection of a steel column. The snapshots presented are not successive computer frames observed during animation. The captured frames are displayed as a filmstrip merely to convey a sense of motion. Video clips of depicted animations are available for download from the first author’s website at http://pathfinder.engin.umich.edu/

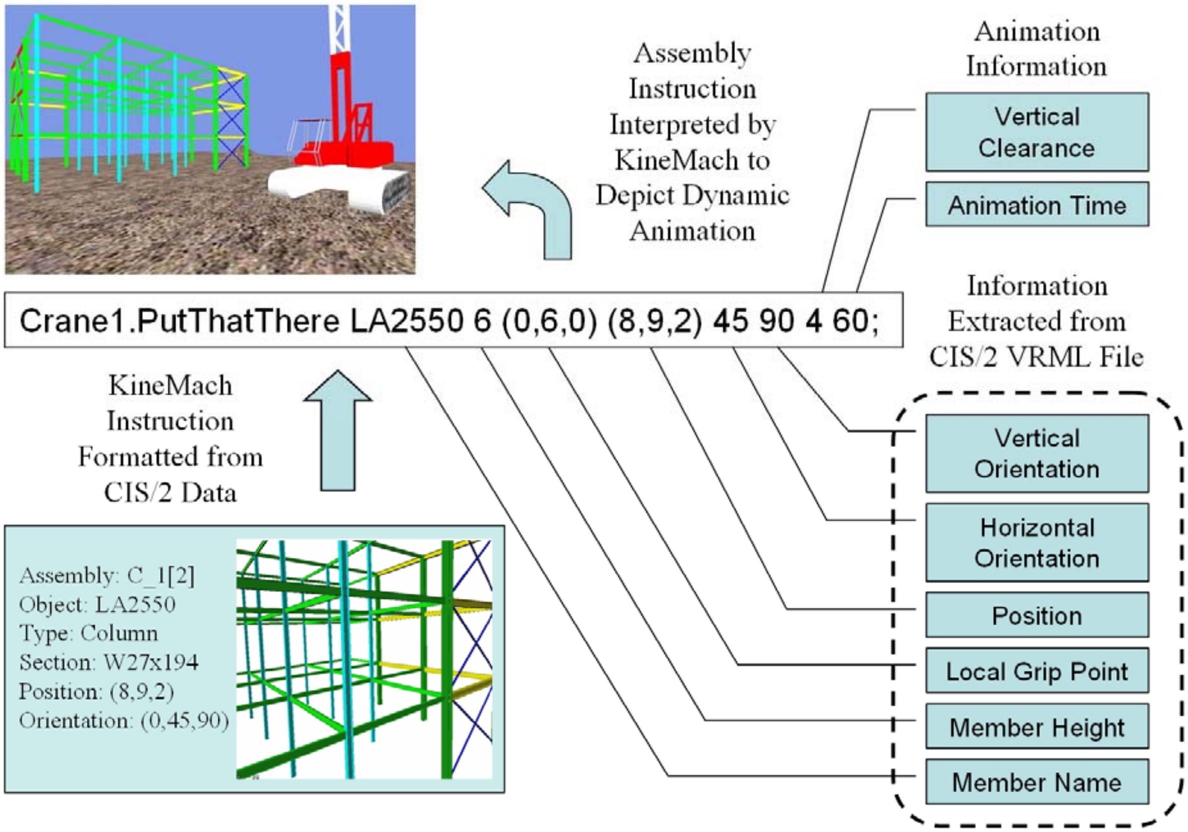


Figure 4: Mapping CIS/2 Geometry and Pose Information to KineMach Arguments

CONCLUSIONS

The presented study investigated the extent to which the CIS/2 standard model can specify product descriptions capable of supporting automated steelwork erection. A kinematically "smart" crane capable of accepting robot-like instructions was implemented in a three-dimensional virtual world. Algorithms to automatically interpret steel member geometry and their spatial configuration from CIS/2 files were designed. The virtual crane was then programmed to use these algorithms to automatically extract steel member information from a CIS/2 file. The extracted information was used to compile automated assembly instructions required to erect the structure inside the virtual world using the crane. Based on the emulation results, it was found that CIS/2 does encapsulate the basic geometry and pose of steel members in a format that, after appropriate geo-referencing, can be readily used to support automated erection of structural steelwork. However, several intermediate information processing steps were found to be necessary before the extracted data can be used to program specific instructions for the automation equipment. In addition, it was found that while CIS/2 itself defines an information rich product model and not intended to have a role in automation, the semantics of many statements could be improved to strengthen the standard's applicability for supporting structural steelwork automation.

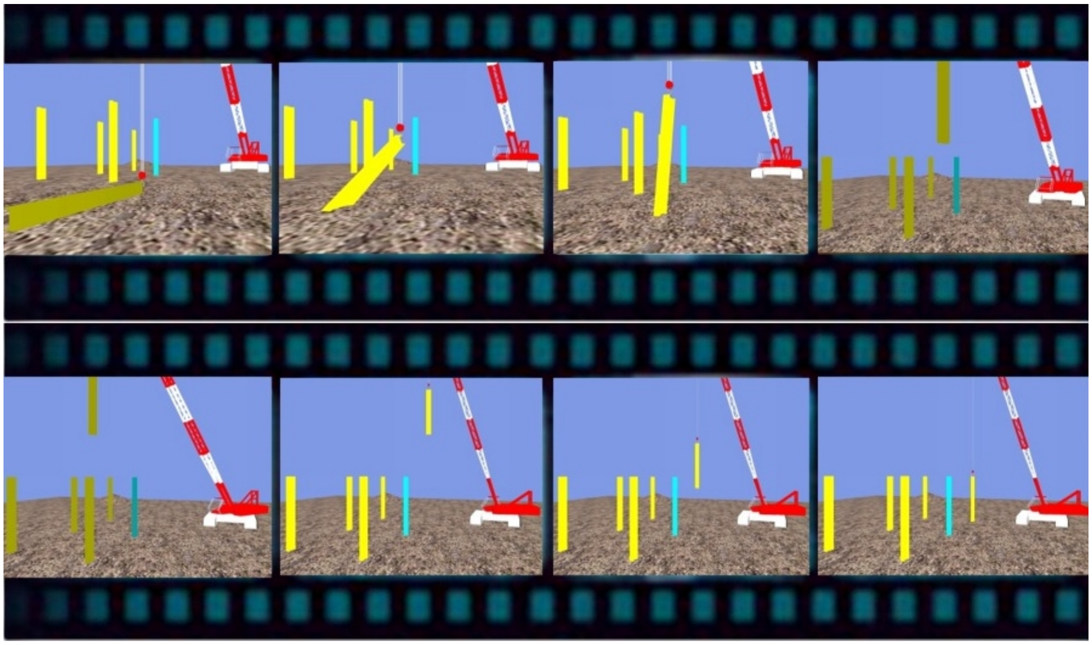


Figure 5: Animation Snapshots of Steel Column Erection

ACKNOWLEDGMENTS

The presented work has been supported by the National Science Foundation (NSF) through grant CMS-0408538. The authors gratefully acknowledge NSF's support. Any opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

Kamat, V. R., and Martinez, J. C. (2005). “Dynamic 3D Visualization of Articulated Construction Equipment”, Journal of Computing in Civil Engineering, Vol. 19, No. 4, American Society of Civil Engineers, Reston, VA.

Lipman, R. R. (2002). “Mobile 3D Visualization for Construction”, 19th Intl. Symposium on Automation and Robotics in Construction (ISARC), NIST, Gaithersburg, MD, 53-58.

Lipman, R.R., K.A. Reed, (2003). “Visualization of Structural Steel Product Models”, Electronic Journal of Information Technology in Construction, Vol. 8, Royal Institute of Technology, Stockholm, Sweden, 43-50.

Lytle, A. M., Saidi, K.S., and Scott, N.A. (2004), “Developments with the NIST Automated Construction Testbed,” 21st International Symposium on Automation and Robotics in Construction (ISARC), September 21-25, 2004, Jeju, Korea.

Reed, K. A. (2002). “Role of the CIMsteel Integration Standards in Automating the Erection and Surveying of Constructional Steelwork”, 19th International Symposium on Automation and Robotics in Construction (ISARC), NIST, Gaithersburg, MD, 15-20.

Saidi, K.S., Lytle, A.M., Scott, N.A. and Stone, W.C. (2005). “Developments in Automated Steel Construction,” NISTIR 7264, National Institute of Standards and Technology, Gaithersburg, MD.