AUTOMATED GENERATION OF DYNAMIC, OPERATIONS LEVEL VIRTUAL CONSTRUCTION SCENARIOS

SUBMITTED: June 2002
REVISED: December 2002
PUBLISHED: May 2003 at http://www.itcon.org/2003/6
EDITOR: Kalle Kahkonen

Vineet R. Kamat, Assistant Professor
Dept. of Civil and Environmental Engineering, University of Michigan, Ann Arbor, Michigan, USA
email: vkamat@engin.umich.edu

Julio C. Martinez, Associate Professor
Dept. of Civil and Environmental Engineering, Virginia Tech, Blacksburg, Virginia, USA
email:julio@vt.edu

SUMMARY*: The use of visualization and virtual reality (VR) technologies to solve operations level problems in construction has been limited to the creation of specific scenarios of very short duration (relatively speaking) resulting from long term efforts dedicated to the creation of specific cases. In order to capitalize on VR technologies in planning and design of construction operations, we must be able to rapidly generate alternate operations level virtual world scenarios for comparison, evaluation, and "what-if" analyses. Several external software and hardware processes are capable of generating information that describes dynamic construction scenarios. However, such processes cannot communicate directly with computer graphics facilities that must be invoked to depict operations in 3D virtual worlds. This paper describes work that led to the design of a specific description that facilitates rapid, automated communication between external authoring processes and 3D computer graphics facilities. This description, formalized as the VITASCOPE language, defines a necessary layer of abstraction that effectively separates 3D virtual construction worlds from the processes that generate them. This is critical in enabling rapid, automated interaction (often simultaneous) between multiple software and hardware processes and 3D virtual worlds.*

KEYWORDS*: 3D Visualization, Construction Operations, Discrete-Event Simulation, Virtual Reality, VITASCOPE.*

1. INTRODUCTION

Planning and control of construction can take place at the project and/or the operation level (Halpin and Riggs 1992). At the project level, a facility is broken down into activities each of which maps to a physical project component (e.g., second floor columns) or to a major time-consuming process (e.g., order and delivery of kitchen cabinets). The planner uses techniques such as the Critical Path Method (CPM) to estimate the time frame during which activities can take place and the times at which important project milestones can be reached.

At the operation level, planning and control are concerned with the technological methods, number and type of resources, and logical strategies required to accomplish an activity or a group of related activities (e.g., erect second floor columns). The effort focuses on work at the field level. The interactions between equipment, labor, materials and space are considered explicitly in the performance of tasks (e.g., lower hook, attach and lift column). The same tasks may be repeated many times, using non-deterministic durations described by probability distributions. Although the planning and control techniques used at the project and operation levels are different, both can benefit substantially from dynamic 3D visualization.

1.1 Project level visualization

Numerous research studies have explored and exploited dynamic 3D visualization at the project level. The efforts are motivated by the shortcomings of traditional scheduling and control techniques such as Bar Charts and CPM in being able to represent all aspects of construction necessary for project level planning (Skolnick

1993). Visualization is achieved by linking a 3D CAD model representing the design of the infrastructure and a construction schedule (Cleveland 1989). This form of visualization has popularly become known as 4D CAD.

4D CAD focuses on the visualization of the construction product over the period of its construction. As time advances, individual components (CAD elements) of the facility are added to the visual model in their final position and form as dictated by the schedule. 4D CAD has been demonstrated to provide construction planners with tangible visual insights that can be of help in planning and controlling construction at the project level (Koo and Fischer 2000). This type of visualization is being exploited because the appropriate technology is straightforward and available. Commensurate with the importance given to project level planning (relative to operations level planning), significant research effort has been invested in improving 4D visualization, both by academia and the industry (e.g. McKinney et. al. 1996, Bentley Systems 2000).

1.2 Operations level visualization

The value of visualizing construction at the operations level is obvious. Being able to visualize construction sequences and operations can result in tremendous savings in money and time and help keep projects on schedule (Alciatore et al. 1991). Visualizing construction operations in 3D can permit the complete subjective analysis of construction processes. Subtle details such as maneuverability problems at loading and dumping areas in earthmoving operations, the restricted visibility of crane operators in steel erection and lifting, overcrowding in work zones due to simultaneous execution of different trades in building construction, and a host of other safety problems such as potential collision between two machines can easily be deciphered by visualizing the actual construction operations that lead to the completion of the constructed product.

In addition, visualizing construction processes in 3D can allow the validation and verification of operational concepts; enable checking for design interferences; and facilitate overall constructability review and the sharing of project information. It can also enable the testing and validation of construction sequences, checking for physical clashes of moving pieces and enable communication/coordination among multiple project participants.

Researchers and industrial proponents of 4D CAD have always been aware of the importance of operations level planning in general and operations level visualization in particular. The awareness of the importance of operations planning and visualization is evidenced by recent 4D CAD research works that aim to convey operations planning information about construction space requirements through 4D visualizations (Riley 1998, Akinci and Fischer 2000). In addition, the need to integrate product and process visualization to encompass both project and operations level planning has also been acknowledged in the literature (Griffis and Sturts 2000).

2. CHALLENGES IN OPERATIONS LEVEL VISUALIZATION

Visualizing construction at the operations level is however a complex proposition that involves being able to view the interaction of the various resources as they build the product or perform a support service. These resources include, but are not limited to, temporary structures, materials, equipment, and labor as they create the product. At this level of detail, visualization of the evolving product can be naturally achieved as a by-product of visualizing the operations that build it.

Visualizing construction operations also encompasses construction procedures that do not necessarily involve the assembly of a tangible product such as a building or a bridge. For instance, construction operations such as paving, tunnelling, quarrying, and earthmoving can obviously be visualized at the operations level. However, at the project level, construction of this nature can only be planned in terms of the desired production rate and has no corresponding visualization (i.e. 4D CAD) context due to the absence of a tangible, laterally limited product that requires assembly.

The planning information that 4D visualization synthesizes is derived from project level planning tools (i.e. CPM schedule and CAD model of the infrastructure). It is not possible to visualize the actual construction operations that lead to the construction of the end product using the sources of 4D CAD (Adjei-Kumi and Retik 1997, Fukai 2000). In other words, 4D visualization can depict the evolution of the construction product but not the interaction of the resources that build it. "True" visualization of construction at the operations level involves being able to "see" graphically on the computer, the operations being carried out in the same way as they would be in the real world. The practical and educational benefits of being able to visualize construction at this level of detail are phenomenal.

In order to visualize an operation it is necessary to see, in addition to the physical components of the facility, the equipment, personnel, materials and temporary structures required to build it. Moreover, it is necessary to depict the movements, transformations and interactions between these visualization elements. The movements and transformations must be spatially and temporally accurate. In order to depict smooth motion, visual elements must be shown at the right position and orientation several times per second. Issues such as trajectories in 3D space, speed and acceleration need to be considered. Due to the amount of detail and precision involved, visualizing construction at this level is a challenging prospect.

3. PREVIOUS RESEARCH AND ITS LIMITATIONS

Many researchers have discussed the potential of 3D operations visualization and virtual interactive environments; some with futuristic perspectives that approach science fiction and that assume that the enabling technology will in time be developed by others. A limited number of researchers have actually experimented with specific scenarios of very short duration (relatively speaking) resulting from long term efforts dedicated to the creation of specific cases (Wakefield and O'Brien 1994, Tsay et al. 1996, Fukuchi et al. 1999, Tseng et al. 2000). Software toolkits (e.g. World Tool Kit) and specific higher-level tools (e.g. 3D Studio, Bentley Dynamic Animator) are available for doing this interactively with the same (significantly large) ratio of 'time invested to develop the specific case' to 'actual duration of the visualization experience'.

Individual developed scenarios and cases can, to an extent, be applied in teaching and training. Such cases however have little applicability in planning and designing construction operations. This is because large scale efforts, a significant amount of time, and sophisticated computing skills, all of which are typically at a premium in a construction setting, are required in developing each individual scenario. In order to effectively apply visualization and virtual reality technologies to construction operations design, we must be able to rapidly and easily generate alternate virtual construction scenarios for comparison and evaluation. This is an absolutely essential feature without which no methodology or tool can be practically used for operations analysis and design.

Conceptually, construction operations can be visualized by linking together discrete-event simulation models and CAD models of the infrastructure as well as of the construction equipment (i.e. machines), temporary structures, and other resources. The result would be 3D animations of simulated (modeled) construction operations. Discrete-event construction process simulation tools facilitate rapid analyses of alternate construction scenarios and provide quantitative guidelines to compare them. However, in order to graphically depict a modeled scenario in a virtual world, such tools must collaborate with appropriate computer graphics facilities (data structures, algorithms and routines). In addition, to cause objects under the control of such simulations to be aware of, and react to, humans and human controlled machines (via hardware controls), the virtual environment (VE) implementation must communicate bi-directionally and at high speed with the simulations, and the controls, each of which could be running in another process and perhaps in another machine.

Discrete-event simulation tools and other processes capable of rapidly generating construction scenarios cannot communicate directly with 3D computer graphics facilities. This is why visualizations depicting construction scenarios have to be either "hard-coded" using high-level languages and graphics libraries or developed interactively using user-interfaces (typically form-based) provided by specific higher-level tools. The large, long-term efforts required in developing such individual cases (some involving interactive hardware controls) renders their use in construction operations design impractical. Additional discussion on previous research and the state-of-the-art in construction operations visualization can be found in (Kamat and Martinez 2001) and (Kamat and Martinez 2002a).

4. POTENTIAL OF CONSTRUCTION OPERATIONS VISUALIZATION

Operations level visualization can be potentially utilized in many ways in construction practice and education. Using available CAD models of the infrastructure and the resources, it can be possible to re-create in a virtual world what happened in the past, what is currently happening (from real-time field data), or what may happen in the future (by showing what was simulated by a discrete-event process simulation model). These visualizations can be very realistic, with accurate depictions of construction sites, infrastructure, equipment, and atmospheric conditions (visibility, fog, rain). Historical (from past data) and predicted (from simulations) animations can be in compressed or expanded time. A 20 second incident could be studied in very slow motion. General operations, in contrast, could be animated in fast motion so that several hours of operations are viewed in a few minutes. By rapidly generating and visualizing construction scenarios in virtual worlds, constructors can study the difference between alternative construction methods, materials, labor levels and management strategies with speed and accuracy at very low cost.

Beyond dynamic 3D animations limited to navigation (i.e., walkthrough), the possibility of interacting and controlling an operation in real time have the potential to add significant value beyond passive visualization as far as learning, operator training, and operations design is concerned. To realize such a vision, and depending on the application, visualization has to be possible on all hardware platforms (Immersive and Non-Immersive). In addition, the visualization application needs to communicate bi-directionally and at high speed with simulation models running in another process and perhaps in another machine. The technology needed to animate operations passively (i.e. walkthrough only) must form the basis for advances that will enable immersive, interactive construction environments to be commonplace. In these virtual environments, objects under the control of simulation models will be aware of, and react to, humans and human controlled machines. The impacts to learning, operator training, and operations design are phenomenal.

As evidenced by many references to work throughout this proposal, construction researchers see tremendous potential in the uses of 3D operations visualization and have significant related questions to investigate. The unavailability of proper information technology has limited the number and value of discoveries that may dramatically reduce the life-cycle costs of constructed facilities.

5. MAIN CONTRIBUTION

Several stages of research are necessary to facilitate rapid automatic generation of alternate construction scenarios (of any duration and complexity) in immersive and interactive 3D virtual worlds. The first of those research stages is the specification and design of an interface that can facilitate communication between computer graphics facilities and external processes capable of automatically describing dynamic construction operations. This interface must define a necessary layer of abstraction that effectively separates visualizations from the processes that generate them. This is essential to facilitate communication between diverse mathematical construction operations analyses tools (e.g. discrete-event simulation systems) and virtual worlds. Such a common interface is also critical if multiple processes and controls are to simultaneously interact with visualizations in real time.

The presented research investigates methods to define such an interface and exploit it to automatically and rapidly generate operations level virtual construction scenarios by facilitating inter-process communication. The work has the following specific objectives:

- Design a specific description that external software (e.g. simulation models) and hardware (e.g. controls) processes can use to communicate a dynamic construction operation.
- Implement a virtual environment application that interprets and processes instructions in that description to recreate a virtual world representation of the communicated operation.
- Validate the effectiveness of the description in facilitating automatic, rapid generation of operations-level virtual construction scenarios using data communicated by external processes.

6. TECHNICAL APPROACH

External processes communicating a dynamic operation in real-time implies that the specific nature of the communication cannot be determined a-priori, as it is dependent on the specific instance of the operation being simulated and/or communicated. Thus, this communication has to be achieved by end-user programming of the driving process (e.g. a discrete-event process simulation model). The transfer of information from external processes to the computer graphics facilities therefore needs to be based on methods that are both expressive (to achieve realistic visualization) and simple (so that they can be generated by end-user programming). The methods have to be open and loosely coupled so that 1) They are independent of any particular driving process (e.g. a specific simulation system) and 2) Processes, other than simulation models, can simultaneously interact with a dynamic visualization.

To achieve the research objectives while addressing these issues, we designed a set of parametric statements that allow external processes to communicate a dynamic visualization. The designed statements together define the specification of a 3D animation language named VITASCOPE (acronym for VIsualizaTion of Simulated Construction OPErations). The VITASCOPE language is capable of describing dynamic construction scenarios by properly illustrating all common construction tasks. VITASCOPE is a straight-line language (Appel 1997). In other words, VITASCOPE language statements that describe a dynamic visualization are only processed sequentially. The language needs no flow control (i.e., while or for loops) because that is achieved by the intended generating processes (e.g., a simulation model while it runs).

We implemented the designed 3D animation language in a corresponding VE application. The VITASCOPE application sequentially 1) Parses and interprets individual animation language statements, 2) Executes appropriate computer graphics algorithms in response to the interpreted statements, and 3) Manages the dynamically evolving CAD database to represent the communicated operation dynamically in a 3D virtual world. To communicate a dynamic visualization, external processes use VITASCOPE's parametric language statements to 1) Control the simulation time and speed, 2) Define multiple motion trajectories that constitute resource movement paths, 3) Create scene objects (terrain, equipment, materials, machines etc.) by instantiating pre-created CAD models, 4) Define behavioral properties for instantiated scene objects, and 5) Manipulate scene objects to describe their dynamic behavior.

Figure 1 presents a schema that describes the relationships between the VITASCOPE language, the corresponding VE application, computer graphics facilities (e.g. algorithms, data structures etc.), and external software and hardware authoring processes. External authoring processes (e.g. simulation models, hardware controls, and real-time positional data streams) are individually and/or collectively capable of generating information that is required for and sufficient to describe dynamic operations level construction scenarios.

The conceptual mechanics of how VITASCOPE automatically converts quantitative information communicated by external processes into dynamic smooth continuous motion are relatively straightforward. As presented in figure 1, The VITASCOPE VE application reads and interprets animation statements (in the syntax of the VITASCOPE language) that describe, chronologically (and sequentially), the things that happen in an operation. In the current VITASCOPE implementation, the animation statements representing static and dynamic events must be pre-recorded sequentially in an ASCII text file (hereinafter referred to as the trace file). The events contained in the trace file can be communicated (in the syntax of the VITASCOPE language) by any external process capable of generating formatted text output as it runs. Such an external process (e.g. a discrete-event process simulation model) can be instrumented (i.e. programmed) to dynamically write the animation trace file while it runs. The communicated VITASCOPE animation statements can refer to 3D CAD objects created with other tools for this purpose. VITASCOPE can read CAD files in many different file formats including VRML. By using the existing CAD models and interpreting the animation instructions communicated by external processes, VITASCOPE recreates a virtual world representation of the described operations. This is achieved by invoking appropriate computer graphics algorithms and routines and applying them to manipulate appropriate CAD models after instantiating them in a 3D virtual world.

VITASCOPE specifies an open description to describe dynamic operations and implements an interpreter that invokes appropriate computer graphics facilities to represent the described operations in a smooth, continuous 3D virtual world. By doing so, VITASCOPE introduces a tangible interface or layer of abstraction that effectively separates dynamic operations visualizations from the processes that can generate them. This has several distinct advantages and opens many opportunities.

For example, loose coupling between process simulation systems and the virtual world environment (i.e. the VITASCOPE application) allows 3D visualizations of operations to be generated from any end-user programmable simulation tool. On the other hand, tight coupling between the VITASCOPE visualization engine and a particular simulation system would compel model developers who desire to visualize their models (created using their system of choice) to learn and use a different simulation tool than the one they are proficient with. In addition, an open and loosely coupled methodology suggests that visualizations need not necessarily be generated by mathematical simulation models. They could also be generated by a wide variety of other external processes. For instance, as figure 1 also presents, real-time data streams (e.g. GPS positional data) could be translated into the syntax of the VITASCOPE language in real-time. This translated stream can then be fed to the VITASCOPE visualization engine to visualize ongoing operations in real-time. In addition, we postulate that the enabling and large scale deployment of immersive, interactive construction virtual environments (discussed in section 4) is directly dependent on, or is greatly facilitated by, enabling a loosely coupled, general-purpose visualization methodology such as that adopted by VITASCOPE.

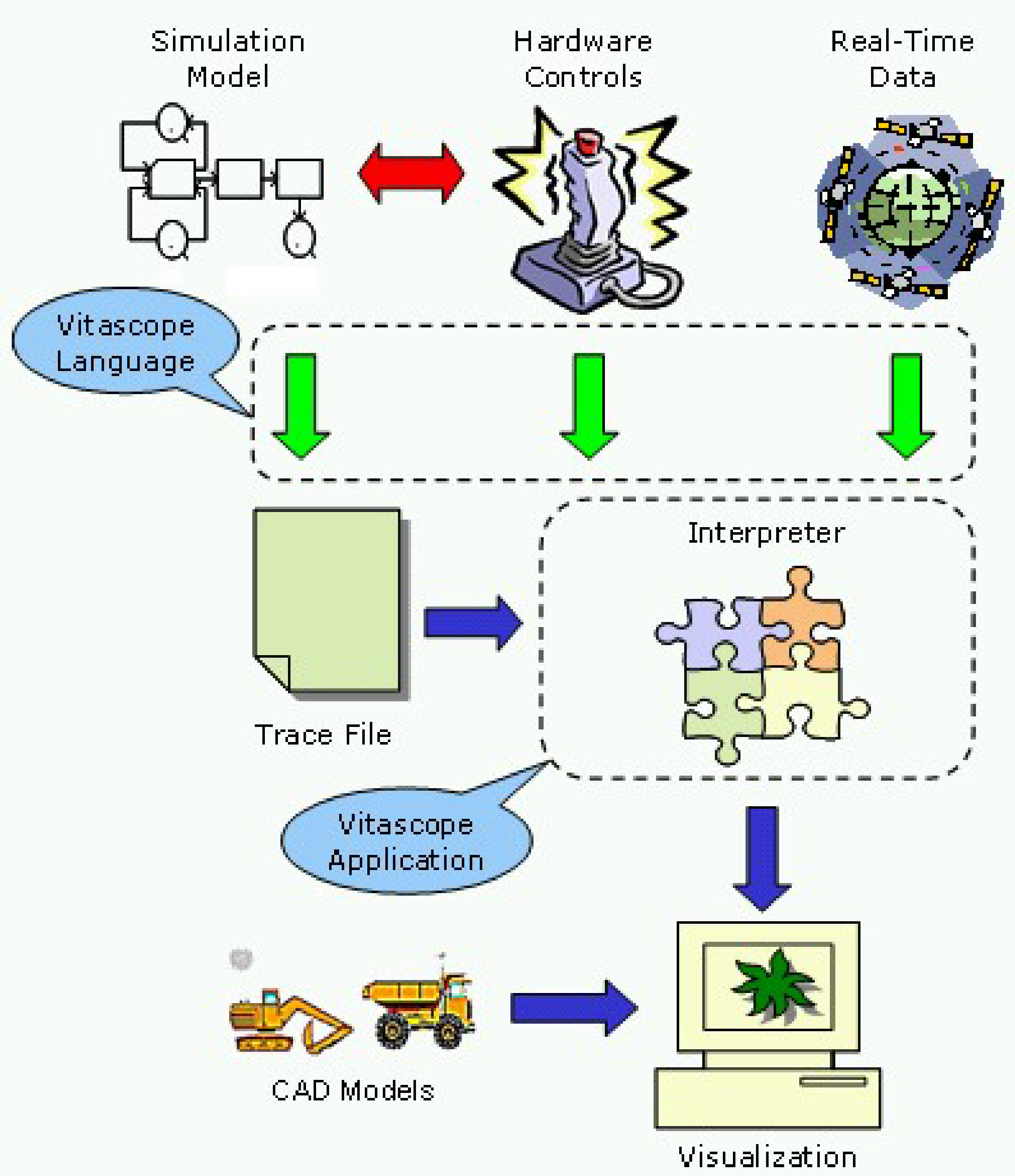

FIG. 1: Relationship between VITASCOPE elements and external authoring processes.

7. THE VITASCOPE LANGUAGE

The VITASCOPE language is a set of 45 parametric statements that together allow external processes to communicate sequential, time-stamped static and dynamic events. These events can describe a smooth and continuous operation of any length and complexity. The communicated events are then interpreted by VITASCOPE's visualization engine. The engine then invokes appropriate data structures, algorithms, and routines to manipulate CAD models and other 3D geometric primitives to present a smooth accurate virtual world representation of the operations communicated by the external processes.

Appendix A presents the VITASCOPE language statements. The statements are grouped according to their functionality as described below. VITASCOPE's parametric language statements are divided into the following functional groups:

7.1 System commands

The major statement in this group (TIME) allows external processes to control the simulation time and specify the instants at which specific events take place (if static) or start (if dynamic). The VIEWRATIO statement allows the animator to interactively control the speed (i.e. viewing ratio) of the visualization, while TIMEJUMP permits instantaneous rewind/fast forward to specific points of interest in a pre-recorded visualization (i.e. a trace file). LOADADDON allows the VITASCOPE VE application to interactively load computer modules that implement extensions to the core VITASCOPE language (Kamat and Martinez 2002c). The final command, SCHEDULE, in this central group complements the TIME statement and allows the execution of other VITASCOPE animation statements to be scheduled at future animation time instants in an animation sequence.

7.2 Scene construction statements

Statements in this group allow external processes to describe the entities (terrain, equipment, materials, machines etc.) that constitute the virtual construction site. This is done by referencing CAD models of relevant resources (e.g. equipment), instantiating (or destroying) multiple specific CAD objects, assembling (or disassembling) CAD objects into logical geometric hierarchies, and placing and orienting objects in the desired state on the virtual construction site. This group also contains statements that are used to define multiple motion paths (i.e. 3D trajectories) that entities will travel on while performing operations.

7.3 Property-setting statements

This group of VITASCOPE statements allows external processes to specify the dynamic behavioral properties of virtual entities that have already been instantiated. An example of such a property is the fore clearance of an object. This property specifies the minimum distance that a trailing object must maintain from another leading object when both are travelling on the same motion trajectory. The rear guide point (RGP) is similarly another property that defines an objects behavior as it travels along curves in a motion path. Statements in this group also allow dynamic strings of text and/or numeric data (e.g. resource properties, operation state variables etc.) to be displayed and dynamically updated in visualizations. To do this, VITASCOPE provides statements that can be used to instantiate text strings, attach them to specific scene objects if necessary, and dynamically update the strings as visualization progresses.

7.4 Dynamic statements

This group of statements constitutes the core of the VITASCOPE language. The group consists of several statements that external processes can use to dynamically manipulate instantiated scene objects to depict the performance of a smooth and continuous operation. Statements in this group describe dynamic geometric transformations of scene objects. These transformations change the positions, orientations, and scales (i.e. sizes) of instantiated objects in the 3D virtual environment to depict the accurate motions objects undergo while performing the communicated operations.

VITASCOPE's parametric animation statements allow an external process to communicate the elemental motions involved in performing construction in a geometric transformation level parlance. VITASCOPE's primary motion-describing statements (e.g. MOVE, ROTATE, SCALE) each describe a single elemental motion that a construction resource undergoes as it performs work (e.g. A truck MOVEs along a haul road, the cab of a backhoe ROTATEs as it swings, a crane's cable SCALEs as the hook is dropped or raised). A time-stamped sequence of an arbitrary number of such elemental motions communicated by an external process is then post-processed to describe a smooth, continuous 3D rendition of the pertinent construction operation.

7.5 View manipulation statements

The last group of VITASCOPE statements allows external processes to programmatically manipulate the position and the orientation of the viewer while in the virtual environment. Statements in this group also allow the user's viewpoint to be attached to (and detached from) a dynamic entity in the visualization. This, for example allows the viewer to ride in the cab of a moving truck or an operating crane. These statements are complemented by the VITASCOPE application's user interface that allows users to navigate to any position and orientation on the virtual jobsite using keyboard keys and/or a mouse to steer.

In appendix A, statement parameters enclosed by < and > signs (e.g. <parameter>) indicate that a relevant numerical value or a text argument is required when feeding the particular statement to the visualization engine. In general, VITASCOPE language statements are quite readable. In addition, describing each statement and its functionality individually is beyond the scope of this paper and is relegated to another relevant document (Kamat and Martinez 2002b).

VITASCOPE's parametric language statements are intended to be simple as well as sufficient. Simplicity, in this context, means that the statements of the language and the information sought by the statement's parameters are both within the authoring capabilities of intended external authoring interfaces such as process simulation systems. The statements are also designed to be sufficient or semantically rich so that the succinct parametric language constructs can encapsulate all information required to describe the performance of construction operations in a smooth and continuous manner. Balancing simplicity and sufficiency (which are conflicting objectives) was a major challenge that was carefully addressed in designing the VITASCOPE language.

8. DESIGNING VITASCOPE

8.1 The appropriate taxonomy

The taxonomy of the construction industry covers a wide range of interests, each appropriate to different people looking at different problems. Any problem in construction requires examination at the appropriate level of detail. In order to identify the appropriate level, the taxonomy of construction must be carefully studied and analyzed from the perspective of the problem that is addressed.

Conceptually, each entity (equipment and human craftsmen) on a virtual construction site can be considered to be equivalent to a robot performing a certain construction task. Just as robots perform construction operations on actual construction sites using real resources, virtual equipment and craftsmen perform construction on the computer screen using virtual resources. The analogy between the two arises from the fact that both robots and virtual entities do not have any intrinsic knowledge about performing construction and need to be instructed (i.e. programmed) to carry out particular construction sequences. In terms of information needs, therefore, virtual construction equipment and human craftsmen are quite similar to construction robots.

We investigated previous works aimed at classifying construction by studying different hierarchical taxonomies developed by various researchers. Many of these taxonomies were developed from interests rooted in automation and robotics (Bernold et. al 1990, Everett and Slocum 1994). Our objective, however, in studying these works was to identify an appropriate level of classifying construction in order to address the problem of designing a description to depict construction operations in 3D virtual worlds.

Both works cited in the paragraph above suggest that classifying construction operations based on basic work tasks (e.g. cut, place, position, connect etc.) is most suitable as far as concentrating automation efforts in construction is concerned. Everett and Slocum (1994) also present a set of 12 basic tasks that are mutually exclusive and collectively encompass all on-site construction operations. The literature also suggests that depending on the nature and the level-of detail of the problem being analyzed, each basic work task can be further broken down into a set of elemental motions (e.g. reach, grasp, put etc.).

8.2 The Geometric Transformation level

Communication with virtual construction entities (e.g. virtual pieces of equipment) can only be achieved using a computer interpretable vocabulary. A virtual piece of equipment cannot be directly told to perform a basic construction task. In fact, such an entity cannot directly be told to perform even elemental motions. In order to communicate instructions in computer interpretable vocabulary, the elemental motion level must be further broken down into geometric transformations such as rotations and translations.

Bernold et al. (1990) implicitly suggest that the geometric transformation level is the next logical level of decomposition (after elemental motions) in the hierarchy of construction field operations. Just as each basic task is comprised of a set of elemental motions, every elemental motion can be broken down into a set of geometric transformations. We concluded that in the context of generally programming virtual construction performers, geometric transformations must be the basic building blocks.

Consider the example of an excavator scooping dirt. Digging can be defined as a basic task (Everett and Slocum 1994). Table 1 breaks down the basic task into a set of elemental motions, which are in turn broken down into geometric transformations. The geometric transformations in the final column are described in the exact syntax of the VITASCOPE language. For instance, the first statement in the third column of table 1 instructs the boom of a virtual excavator to rotate in the vertical plane (indicated by VERT) by an amount of 27 degrees in the clockwise direction (indicated by the negative sign) in 12 time units (seconds in this case). The statement in the fourth row similarly instructs the cab of the virtual excavator to rotate in the horizontal plane (indicated by HOR) by an amount of 90 degrees in the anticlockwise direction in 16 seconds. We can communicate construction operations to virtual pieces of equipment only at this geometric transformation level.

Table 1: Geometric Transformation level decomposition.

Basic Task	**Elemental Motions**	**Geometric Transformations**
Dig Dirt	Lower Empty	ROTATE Boom VERT -27 12; ROTATE Stick VERT -10 9;
	Scoop Dirt	ROTATE Bucket VERT -90 12;
	Lift Loaded	ROTATE Boom VERT 27 16; ROTATE Stick VERT 10 12;
	Swing Loaded	ROTATE Cab HOR 90 16;
	Lower Loaded	ROTATE Boom VERT -35 10; ROTATE Stick VERT 20 8;
	Dump	ROTATE Bucket VERT 90 6;
	Lift Empty	ROTATE Stick VERT -20 12; ROTATE Boom VERT 35 9;
	Swing Empty	ROTATE Cab HOR -90 14;
	Lower Empty	ROTATE Boom VERT -27 12; ROTATE Stick VERT -10 9;

The motions of most construction equipment and craftsmen can ultimately be broken down into rotations, translations, and other geometric transformations. This is the only level at which instructions to perform construction sequences can be directly communicated to virtual pieces of equipment and craftsmen. The basic dynamic constructs of the VITASCOPE language are therefore designed at the geometric transformation level.

8.3 Parameterization

Many parts of construction work have been identified as repetitive and cyclic. However, while each construction work cycle might be similar to its predecessor and successor, the cycles are not identical (Everett and Slocum 1994). The basic tasks comprising work cycles, although structurally similar, are different in amplitude and direction from their predecessors and successors.

For example, the excavator scooping dirt goes through the same set of motions in loading a truck i.e. dig, lift, swing loaded, dump, swing empty, and dig again. However, the amplitude of these motions may vary for each pass of the excavator. This could either be due to the increasing depth of the hole that is being dug, or the repositioning of the excavator as it performs work, or the fact that each successive empty truck does not stop at exactly the same spot.

Therefore, although the performance of construction can ultimately be broken down into geometric transformations, their amplitudes are not constant and cannot be predetermined. Complexity further increases with the increase in the degrees of freedom each piece of equipment or a human craftsman has and exercises while performing construction.

VITASCOPE addresses this problem by specifying parameterized instructions. In table 1, the elemental motions involved in performing the basic task of digging have been decomposed into basic geometric transformations (i.e. rotations). However, the direction of rotation (i.e. plane of rotation) is different for each elemental motion. For example, the cab of an excavator rotates on a plane that is parallel to the surface on which the excavator rests. The boom, stick, and the bucket, however, rotate along a plane that is perpendicular to the resting surface and parallel to the forward direction of the cab. In addition, the amplitude of rotations during each elemental motion differs as the excavator adjusts to the deforming terrain and due to other factors.

In VITASCOPE, this issue is addressed by designing an ability to specify parametric instructions to virtual pieces of equipment and craftsmen. For example, in the third column of table 1, the third and fourth arguments in the rotation instructions specify the plane and the amplitude of rotation for each of the geometric transformations that comprise the elemental motions. The fifth argument specifies the time required for the particular instant of the transformation and will obviously be different each time. In general, all numerical parameters are typically described by sampling values from probability distributions.

9. THE VITASCOPE VIRTUAL ENVIRONMENT APPLICATION

The VITASCOPE application is the VE implementation that interprets and processes sequential instructions (in the syntax of the VITASCOPE language) to recreate a 3D virtual world representation of operations communicated by external processes. In order to facilitate communication between external authoring processes and computer graphics facilities, the application implements three key technologies: 1) The conversion of discrete animation information (i.e. parametric VITASCOPE language statements) into smooth motion; 2) The spatial organization and rendering of multiple dynamic 3D CAD objects; and 3) The efficient sequencing and timing of frames such that the ratio of viewing to simulated time is constant.

The VITASCOPE VE application requires a graphical database of 3D scene objects that must be created, manipulated, and maintained in order to depict animation. Scene graph architectures are effective for organizing such databases and are well supported by several industrial-strength commercial libraries (Kamat and Martinez 2002a). The dynamic maintenance of scene graphs needed to represent virtual construction worlds that are constantly evolving, however, requires the development of algorithms specific to the application. The VITASCOPE VE application creates, manipulates, and manages such scene graph based databases to animate communicated construction operations. The scene graph based data structures and algorithms designed in implementing the VITASCOPE VE application are interesting research results that are discussed in detail in (Kamat and Martinez 2002a).

10. VALIDATION

We validated the effectiveness of the designed specification (i.e. the VITASCOPE language) and its implementation (i.e. the VITASCOPE application) in facilitating automatic, rapid generation of operations-level virtual construction scenarios communicated by external processes. To do this, we instrumented discrete-event process simulation models to automatically generate animation trace files (in the VITASCOPE language) as they run. The instrumented models were then executed to produce the required animation trace files. A simulation model typically executes in only a few seconds even though the animation trace file it produces is typically several thousand lines long and may describe a VITASCOPE visualization spanning minutes, hours, or even years.

We then post-processed the trace files in VITASCOPE's VE application to investigate the extent to which the simulated (and communicated) operations can be recreated in 3D virtual worlds. The VITASCOPE application was able to accurately depict graphical 3D representations of the communicated operations. The degree of accuracy was faithful to the amount of detail communicated by the driving process (i.e. the discrete event simulation models in this case). In order to generate alternate virtual scenarios of each operation, we then simply manipulate the quantitative and/or logical simulation parameters of the driving process model. These parameters include the number and type of resources (e.g. number of trucks in an earthmoving operation, type of crane to use in steel erection, space for temporary storage of materials etc.); the rules under which the different tasks that compose the operations are performed; and different random variates to describe the durations of individual tasks. The modified model is then promptly rerun to produce an alternate virtual world scenario that can then be processed and visualized by VITASCOPE for evaluation. This procedure of instrumenting a discrete-event process model and manipulating it's parameters to generate a VITASCOPE virtual world scenario is described in detail with the help of a worked example in (Kamat and Martinez 2001).

Figure 2 presents snapshots of some construction operations that were modeled, communicated, and visualized as part of VITASCOPE's validation exercise. The types of construction operations that have been visualized include earthmoving, block masonry, delivery and placement of concrete, and steel erection.

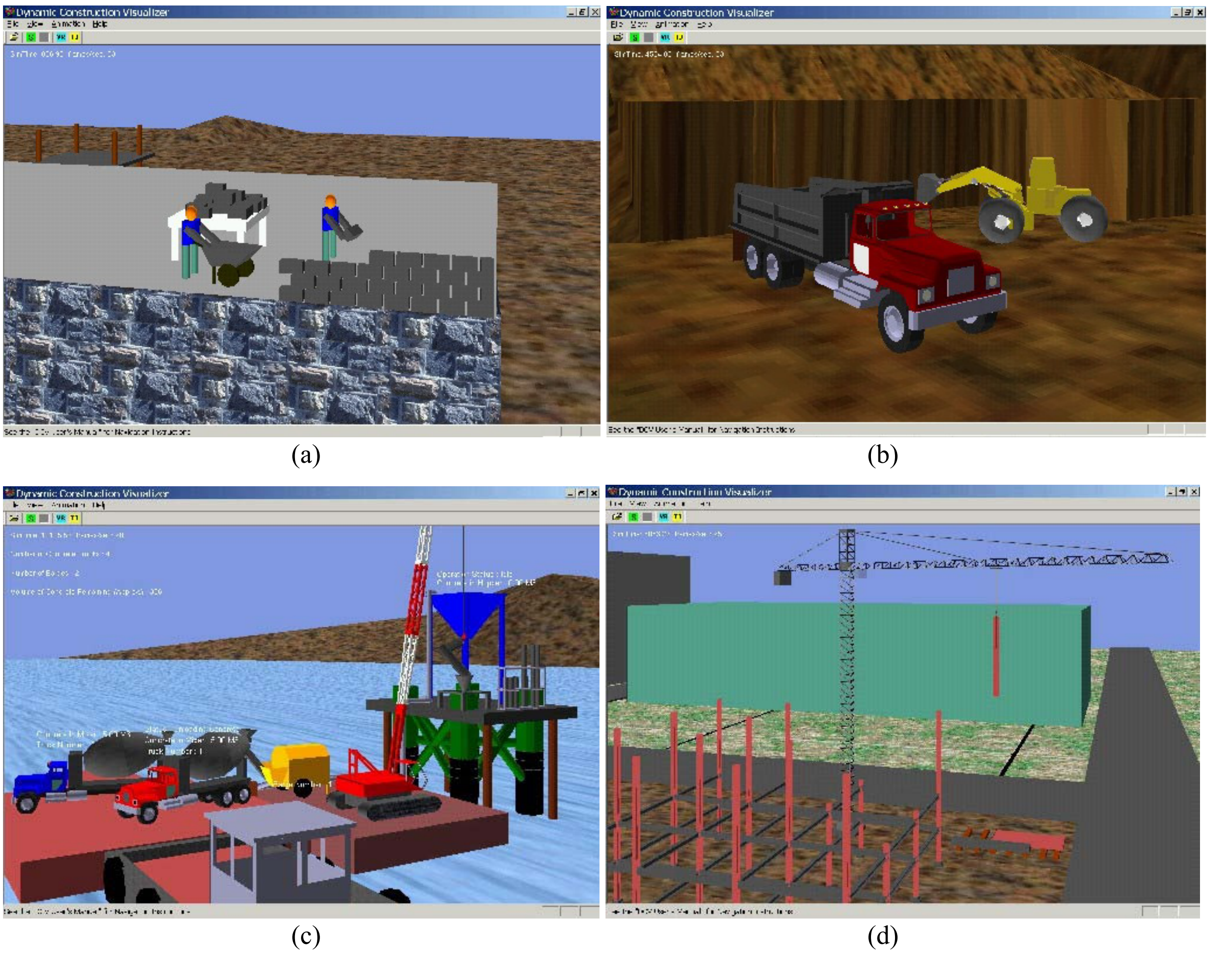

FIG 2: Snapshots of visualized operations.

Appendix B includes a short portion of the VITASCOPE animation trace file that describes the concrete delivery and pouring operation presented in figure 2(c). The entire VITASCOPE animation trace files and the driving discrete-event process models are both significantly large to include in this paper even as appendices. However, complete animation trace files and the driving simulation process models are available for download and inspection from this paper's accompanying website (http://strobos.cee.vt.edu/vitascope/ITCON). Files of both types can be inspected by opening them in any formatted text editor. The accompanying website also hosts demonstration movies of each of the VITASCOPE visualizations whose snapshot is presented in figure 2.

11. EXTENSIBILITY

As the animation trace files and the corresponding videos prove, most elemental motions involved in performing common construction operations can be ultimately broken down into parametric geometric transformations. However, the number of such transformations required for even a simple set of elemental motions can be very large. For example, as presented in table 1, communicating just one pass of an excavator that has only two possible planes and four axes of rotation, requires the specification of 14 geometric transformations. It is left to the reader to imagine how many geometric transformations might be required to realistically depict a human craftsman having several joints (axes of rotation) and infinite degrees of freedom (planes of rotation and translation).

We envision that elemental motions and even basic tasks (in some cases) can be communicated to virtual construction resources using higher-order language constructs. These higher-order constructs can concatenate basic geometric transformations such as rotations and translations to describe elemental motions or even basic tasks. The parametric components will still remain, although their number and the values they represent may indicate factors other than those required in specifying basic geometric transformations. The VITASCOPE language and the corresponding VE application are therefore designed to be extensible and scalable. Researchers and engineers can design extensions to the VITASCOPE language and implement them without having to understand or modify the existing implementation (Kamat and Martinez 2002c).

Revisiting the digging excavator example discussed in section 8.2, it can be possible to describe the elemental motions involved in a pass using just one parametric statement as described in figure 3. The language construct EXCAVATORPASS is described as being comprised of various geometric transformations (i.e. rotations). The parameters required in communicating the pass to a virtual excavator are now the location of the dirt and the position of the empty truck and not the values and planes of individual rotations. Figure 3 is an oversimplified version provided to clearly present the concept of scalability and extensibility. A true higher-order definition for such a pass will obviously also include a representation of time both inside the function body as well as in the list of parameters (Kamat and Martinez 2002d).

```
EXCAVATORPASS (DirtLocation, TruckLocation)
{
      ROTATE Boom VERT -27;
      ROTATE Stick VERT -10;
      ROTATE Bucket VERT -90;
      ROTATE Boom VERT 27;
      ROTATE Stick VERT 10;
      ROTATE Cab HOR 90;
      ROTATE Boom VERT -35;
      ROTATE Stick VERT 20;
      ROTATE Bucket VERT 90;
      ROTATE Stick VERT -20;
      ROTATE Boom VERT 35;
      ROTATE Cab HOR -90;
      ROTATE Boom VERT -27;
      ROTATE Stick VERT -10;
}
```

FIG 3: Designing higher-order language constructs.

Combinations of elementary geometric transformations could similarly be used to describe other elemental motions or even basic tasks if they can be encapsulated by a finite number of parameters. Describing the performance of all common construction operations, especially the accurate motions of equipment and human craftsmen, using only basic geometric transformations is extremely cumbersome. On the other hand, designing higher-level constructs to communicate performance of all elemental motions and basic tasks is challenging, arduous, time-consuming, and beyond the capabilities of a single researcher. Extensibility and scalability of the described nature is therefore essential if the VITASCOPE language is to evolve substantially over time and through the collective efforts of many research scientists.

12. FUTURE WORK

Statements in the VITASCOPE language describe the performance of a construction process as a concatenation of elemental motions in geometric transformation level dialect. This parlance is advantageous for its generality and flexibility. However, it is tedious and often impossible to realistically describe the motions of certain, highly articulated construction resources using this low-level vocabulary. For instance, describing the realistic motions construction workers undergo as they perform work is a challenging proposition using VITASCOPE's elemental motion based language. Additional research is necessary to design and implement higher level animation methods that can succinctly describe the complex motions of articulated resources in a finite number of parametric text statements.

VITASCOPE's efficacy has been validated using discrete-event simulation models as external authoring interfaces in post-processed mode. While this exercise confirms the simplicity and expressiveness of the designed animation language, the ability to generate a smooth, continuous, 3D animation using other software and hardware authoring interfaces needs to be investigated. For instance, exploring the possibility of using real-time data streams (e.g. from GPS receivers mounted on equipment pieces) to describe a virtual world representing the current status of a jobsite presents an interesting research initiative. In addition, integration of simulation models with hardware controls to describe interactive virtual construction worlds requires the investigation of several research issues and calls for some interesting work.

13. CONCLUSION

In order to effectively apply virtual reality technologies to construction operations design, we must be able to rapidly generate alternate operations-level virtual world scenarios for comparison, evaluation, and "what-if" analyses. To do this, different external processes capable of describing construction operations must be able to interact (often simultaneously) with 3D virtual environments through a common description. The interface that such a description defines must effectively separate visualizations from the processes that generate them.

A general description to communicate dynamic construction operations of any length and complexity must be carefully designed at the correct level of abstraction. Previous works identify the basic task as the fundamental building block of all construction field operations. This study observed that the basic task level, and indeed the next elemental motion level, are both too general as far as communicating construction sequences to virtual pieces of equipment and human craftsmen is concerned. A taxonomical level based on basic geometric transformations is therefore identified as appropriate for the design of such a description and subsequently used in designing the VITASCOPE 3D animation language.

Validation of the designed VITASCOPE language effectively proves that the geometric transformation level is not only practical in allowing external processes to communicate dynamic construction operations, but also quite effective. In addition, by post-processing simulation model-generated trace files to recreate dynamic construction operations, we are able to conclude that the designed interface effectively separates and entirely decouples the virtual world (i.e. the visualization engine) from the processes that generate dynamic scenarios. The validation exercise also confirms that VITASCOPE facilitates rapid generation and visualization of alternate operations level construction scenarios. This is achieved by changing decision variables in discrete-event process models and communicating the operations to the VE implementation in the language's syntax.

14. ACKNOWLEDGMENTS

We thank the National Science Foundation (CAREER and ITR programs) for supporting the presented work. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

15. REFERENCES

Adjei-Kumi, T., and Retik, A. (1997). Library-based 4D Visualization of Construction Processes, Proceedings of the 1997 IEEE Information Visualization Conference, IEEE, Piscataway, New Jersey, USA, pp. 315-231.

Akinci, B., and Fischer, M. (2000). An Automated Approach for Accounting for Spaces Required by Construction Activities, Proceedings of the Sixth Construction Congress, American Society of Civil Engineers, Reston, Virginia, USA, pp. 1-10.

Alciatore, D.G., O'Connor, J.T., and Dharwadkar, P. (1991). A Survey of Graphical Simulation in Construction: Software, Usage, and Application, CII Source Document 68, Construction Industry Institute, The University of Texas at Austin, Austin, Texas, USA.

Appel, A.W. (1997). Modern Compiler Implementation in C: Basic Techniques, Cambridge University Press, Cambridge, United Kingdom.

Bentley Systems (2000). Bentley Schedule Simulator, Bentley Systems, Exton, Pennsylvania, USA.

Bernold, L. E., Abraham, D.M., and Reinhart, D.B. (1990). FMS Approach to Construction Automation, Journal of Aerospace Engineering, 3(2), American Society of Civil Engineers, Reston, Virginia, USA, pp 108-121.

Cleveland, A.B. Jr. (1989). Real-Time Animation of Construction Activities, Proceedings of the 1st Construction Congress, American Society of Civil Engineers, Reston, Virginia, USA, pp. 238-243.

Everett, J.G., and Slocum, A.H. (1994). Automation and Robotics Opportunities: Construction versus Manufacturing, Journal of Construction Engineering and Management, 120(2), American Society of Civil Engineers, Reston, Virginia, USA, pp. 443-452.

Fukai, D. (2000). Beyond Sphereland: 4D-CAD in Construction Communications, Proceedings of the Sixth Construction Congress, American Society of Civil Engineers, Reston, Virginia, USA, pp. 1001-1006.

Fukuchi, Y., Hirai, Y., Kobayashi, I., Hoshino, Y., and Kazuhiro, Y. (1999). Application of Collaborative Supported Frame Accurate Animation for Bridge Construction Project, Proceedings of the IEEE International Conference on Information Visualization , Piscataway, New Jersey, USA, pp. 166-173.

Griffis, F.H., and Sturts, C.S. (2000). FIAPP and the Three-Dimensional Computer Model, Proceedings of the Sixth Construction Congress, American Society of Civil Engineers, Reston, Virginia, USA, pp. 996-1000.

Halpin, D.W., and Riggs, L.S. (1992). Planning and Analysis of Construction Operations, John Wiley and Sons, New York, USA.

Kamat, V. R., and Martinez, J. C. (2001). Visualizing Simulated Construction Operations in 3D, Journal of Computing in Civil Engineering, Vol. 15, No. 4, American Society of Civil Engineers, Reston, Virginia, USA, pp. 329-337.

Kamat, V. R., and Martinez, J. C. (2002a). Scene Graph and Frame Update Algorithms for Smooth and Scalable 3D Visualization of Simulated Construction Operations, Journal of Computer-Aided Civil and Infrastructure Engineering, Vol. 17, No. 4, Blackwell Publishers, Malden, Massachusetts, USA, pp. 228-245.

Kamat, V. R., and Martinez, J. C. (2002b). The VITASCOPE Language Reference Manual, Working document, Virginia Tech, Blacksburg, Virginia, USA.

Kamat, V. R., and Martinez, J. C. (2002c). "Mechanisms for Extensible and Scalable 3D Visualization of Construction Processes and Products", Technical Report, Vecellio Construction Engineering and

Management Program, Virginia Tech, Blacksburg, VA. Available: <http://filebox.vt.edu/users/vkamat/ExtensibleWP.pdf > (Dec. 20, 2002).

Kamat, V.R., and Martinez, J.C. (2002d). “Practical 3D Animation of Simulated Construction Operations Involving Multiply Articulated Equipment”, Technical Report, Vecellio Construction Engineering and Management Program, Virginia Tech, Blacksburg, VA. Available: <http://filebox.vt.edu/users/vkamat/PracticalWP.pdf> (Dec. 20, 2002).

Koo, B., and Fischer, M. (2000). Feasibility Study of 4D CAD in Commercial Construction, Journal of Construction Engineering and Management, Vol. 126, No. 4, American Society of Civil Engineers, Reston, Virginia, USA, pp. 251-260.

McKinney K., Kim J., Fischer M., and Howard C. (1996). Interactive 4D-CAD, Proceedings of the 3rd Congress on Computing in Civil Engineering, American Society of Civil Engineers, Reston, Virginia, USA, pp. 383-389.

Riley, D.R. (1998). 4D Space Planning Specification Development for Construction Work Spaces, Proceedings of the International Congress on Computing in Civil Engineering, American Society of Civil Engineers, Reston, Virginia, USA, pp. 354-363.

Skolnick, J.F. (1993). A CAD-Based Construction Simulation Tool Kit for Construction Planning, Proceedings of the Fifth International Conference (V-ICCCBE) sponsored by the Technical Council on Computer Practices of the American Society of Civil Engineers, Reston, Virginia, USA, pp. 117-124.

Tsay, T.C., Hadipriono, F.C., and Larew, R.E. (1996). Virtual Reality Modeling for Bridge Construction, Proceedings of the 3rd Congress on Computing in Civil Engineering, American Society of Civil Engineers, Reston, Virginia, USA, pp. 63-69.

Tseng, C.H., Hadipriono, F.C., Duane, J., and Larew, R.E. (2000). A Three Dimensional Construction Operation Simulation of a Multistory Building in a Virtual Environment, Dept. of Civil and Environmental Engineering and Geodetic Science, Ohio State University, Columbus, Ohio, USA. (Available at: http://www.siu.edu/~coalctr/paper393.htm)

Wakefield, R.R., and O’Brien, J.B. (1994). Real-Time, Visual Output Simulation – A New Process Tool for Construction Planners and Managers, Proceedings of the National Construction and Management Conference on Recent Advances in Construction and Management, Balkema, Rotterdam, Netherlands, pp. 71-86.

APPENDIX A: VITASCOPE LANGUAGE SET

1. SYSTEM COMMANDS

```
TIME <EventTimeValue>;
TIMEJUMP <TimeValue>;
VIEWRATIO <ViewRatioValue>;
LOADADDON <AddOnName>;
SCHEDULE <AnimationStatement> <ExecutionTime>;
END;
/ <Comment>
```

2. SCENE CONSTRUCTION STATEMENTS

```
CLASS <ClassName> <CADFileName>;
ORIENTCLASS <ClassName> <AboutAxis> <RotationAmount>;
CREATE <ObjectName> <ClassName>;
DESTROY <ObjectName>;
ATTACH <ChildObjectName> <ParentObjectName> <AttachPoint>;
ATTACHNOSCALE <ChildObjectName> <ParentObjectName> <AttachPoint>;
DETACH <ChildObjectName>;
CHANGECLASS <ObjectName> <ClassName>;
PLACE <ObjectName> AT <PlacePoint>;
PLACE <ObjectName> ON <PathName>;
HORIZORIENT <ObjectName> <RotationValue>;
VERTORIENT <ObjectName> <RotationValue>;
PATH <PathName> <Points>;
NONDIRECPATH <PathName> <Points>;
```

3. PROPERTY-SETTING STATEMENTS

```
SET CLASS <ClassName> RGP <Value>;
SET CLASS <ClassName> FORECLEARANCE <Value>;
SET CLASS <ClassName> AFTCLEARANCE <Value>;
SET OBJECT <ObjectName> RGP <Value>;
SET OBJECT <ObjectName> FORECLEARANCE <Value>;
SET OBJECT <ObjectName> AFTCLEARANCE <Value>;
OBJECTSTAT <ObjectStatName> <InitialString>;
```

```
ATTACHSTAT <ObjectStatName> <ParentObjectName> <AttachPoint>;
UPDATEOBJSTAT <ObjectStatName> <NewString>;
OPERATIONSTAT <OperationStatName> <InitialString>;
UPDATEOPERSTAT <OperationStatName> <NewString>;
```

4. DYNAMIC STATEMENTS

```
MOVE <ObjectName> <PathName> <TravelDuration>;
MOVESPEED <ObjectName> <PathName> <TravelSpeed>;
SLIDE <ObjectName> <TranslationValue> <TravelDuration>;
TGTSLIDE <ObjectName> <TargetTranslationValue> <TravelDuration>;
ROTATE <ObjectName> HOR <RotationAmount> <RotateDuration>;
TGTROTATE <ObjectName> HOR <TargetRotation> <RotateDuration>;
ROTATE <ObjectName> VERT <RotationAmount> <RotateDuration>;
TGTROTATE <ObjectName> VERT <TargetRotation> <RotateDuration>;
SCALE <ObjectName> <ScaleFactor> <ScaleDuration>;
TGTSCALE <ObjectName> <TargetScaleFactor> <ScaleDuration>;
```

5. VIEW MANIPULATION STATEMENTS

```
VIEWPOINT <Name> <Position> <OrientationAxis> <RotationAmount>;
ATTACHCAMERATO <ObjectName> <AttachPoint>;
DETACHCAMERAFROM <ObjectName>;
CAMERA NEARCLIPPLANE <Value>;
CAMERA FARCLIPPLANE <Value>;
```

APPENDIX B: PORTION OF VITASCOPE ANIMATION TRACE FILE

```
/ Define all motion paths
PATH  ConcreteTruckBackUpToUnLoad         '(-0.74,3,-44.2)'
                                          '(2.64,3,-36.1)';
... Define all other trajectories

TIME 0;

/ Position and orient camera
VIEWPOINT View1 '(-47.83, 16.74, 112.48)' '(0,1,0)' 1.51;

/ Define all object classes (i.e. CAD files)
/ The mobile crane
CLASS CraneCrawler      Crawler.wrl;
CLASS CraneBoom         Boom.wrl;
CLASS CraneCabin        Cabin.wrl;
CLASS CraneCable        Cable.wrl;
CLASS CraneHook         Hook.wrl;
... Define all other classes

/ Create, assemble, and place objects in scene
CREATE Crawler1   CraneCrawler;
CREATE Cabin1     CraneCabin;
ATTACH Cabin1     Crawler1    (0,2,0);
CREATE Boom1      CraneBoom;
ATTACH Boom1      Cabin1      (0,0,0);
CREATE Cable1     CraneCable;
ATTACH Cable1     Boom1       (0,55,0);
CREATE Hook1      CraneHook;
ATTACHNOSCALE     Hook1 Cable1      (0,-1,0);
PLACE Crawler1 AT (-5,2.9,-15);
... Create, assemble, and place all other objects

... Static and dynamic time stamped events
TIME 8914.30;
TGTROTATE Cabin1 HOR 270.00 26.57;
TGTROTATE Cable1 VERT 10.47 26.57;
TGTROTATE Boom1 VERT -10.47 26.57;
TIME 8940.87;
TGTSCALE Cable1 (1,44.48,1) 25.52;
TGTSLIDE Hook1 (0,-44.48,0) 25.52;
TIME 8994.52;
ATTACH TremieFunnel Hook1 (0,-0.6,0);
TIME 8994.52;
SCALE Cable1 (0,-5.00,0) 26.86;
SLIDE Hook1 (0,5.00,0) 26.86;
TIME 9021.38;
TGTROTATE Cabin1 HOR 296.57 29.37;
TGTROTATE Cable1 VERT 11.73 29.37;
TGTROTATE Boom1 VERT -11.73 29.37;
TIME 9027.62;
MOVE Truck2 TruckEntersBarge 50.60;
TIME 9050.75;
```

```
TGTSCALE Cable1 (1,46.05,1) 26.66;
TGTSLIDE Hook1 (0,-46.05,0) 26.66;
TIME 9078.22;
ATTACH Truck2 Barge1 (16.92,3,-6.42);
TIME 9078.22;
MOVE Barge1 BargeBacksUp 109.29;
TIME 9104.79;
DETACH TremieFunnel;
PLACE TremieFunnel AT (0,12.15,-5);
TIME 9114.79;
SCALE Cable1 (0,-5.00,0) 25.87;
SLIDE Hook1 (0,5.00,0) 25.87;
TIME 9140.65;
TGTROTATE Cabin1 HOR 270.00 26.97;
TGTROTATE Cable1 VERT 10.47 26.97;
TGTROTATE Boom1 VERT -10.47 26.97;
TIME 9187.51;
MOVE Barge1 BargeTravelsToPier 583.19;
TIME 9187.51;
MOVE Barge2 EmptyBargeDocksAtShore 197.63;
TIME 9194.73;
TGTSCALE Cable1 (1,44.48,1) 27.68;
TGTSLIDE Hook1 (0,-44.48,0) 27.68;
TIME 9250.87;
ATTACH TremiePipe2 Hook1 (0,-0.6,0);
TIME 9250.87;
SCALE Cable1 (0,-6.66,0) 28.56;
SLIDE Hook1 (0,6.66,0) 28.56;
TIME 9308.00;
SCALE Cable1 (0,-6.34,0) 13.53;
SLIDE Hook1 (0,6.34,0) 13.53;
TIME 9308.00;
TGTROTATE Cabin1 HOR 257.07 27.06;
TGTROTATE Cable1 VERT 16.53 27.06;
TGTROTATE Boom1 VERT -16.53 27.06;
TIME 9335.06;
TGTSCALE Cable1 (1,38.27,1) 25.94;
TGTSLIDE Hook1 (0,-38.27,0) 25.94;
TIME 9385.14;
DETACH Truck3;
TIME 9385.14;
MOVE Truck3 ConcreteTruckBackUpToExit 5.63;
TIME 9388.80;
DETACH TremiePipe2;
PLACE TremiePipe2 AT (-8.5,18.81,0.25)
TIME 9390.77;
MOVE Truck3 ConcreteTruckExitBarge 40.85;
TIME 9398.80;
SCALE Cable1 (0,-5.00,0) 28.80;
SLIDE Hook1 (0,5.00,0) 28.80;
TIME 9427.60;
TGTROTATE Cabin1 HOR 270.00 29.49;
TGTROTATE Cable1 VERT 10.47 29.49;
TGTROTATE Boom1 VERT -10.47 29.49;
```

```
TIME 9431.61;
DETACH Truck4;

... More static and dynamic events
```