N96- 14993

Spatial Considerations for Instructional Development in a Virtual Environment

Laurie McCarthy
Michael Pontecorvo
Frances Grant
Randy Stiles

Lockheed Palo Alto Research Laboratories*

Abstract

In this paper we discuss spatial considerations for instructional development in a virtual environment. For both the instructional developer and the student, the important spatial criteria are perspective, orientation, scale, level of visual detail, and granularity of simulation. Developing a representation that allows an instructional developer to specify spatial criteria and enables intelligent agents to reason about a given instructional problem is of paramount importance to the success of instruction delivered in a virtual environment, especially one that supports dynamic exploration or spans more than one scale of operation.

1.0 Introduction

The motivation for integrating an instructional system with a simulation is usually to support exploratory learning. In the case of a virtual environment for instruction, the main purpose is to give the students three-dimensional visual feedback during exploration. Getting the proper mix of exploration and visual presentation can be difficult. In an exploratory setting, an instructional developer normally does know ahead of time if the student will be able to witness certain events in the simulation. Our goal is to support the developer's task of explanation by providing a representational system that can be used to support reasoning about the appropriate spatial presentation of events, and a set of visual abstractions for manipulating the underlying representation.

The spatial issues presented in this paper are an outgrowth of the research conducted for the design of Lockheed's virtual environment instructional system. We are working on a visual programming system to support the instructional developer in the seamless visual development of lesson plans, objects, and simulations situated in a virtual environment (Stiles, 1992). The underlying metaphor for the Lockheed system is that of a television studio, with a studio control booth, stage, and audience section (see Figure 1). The control booth serves as the developer's information workspace (Card, 1991), providing all the tools required for courseware development. The visual simulation and interactions with the system are carried out on the studio stage, where the trainee may participate and affect the outcome of a given instructional simulation. The audience metaphor allows passive observation, and if the instructional developer allows it, provides the trainee the freedom of movement within the virtual environment without affecting the simulation (Grant, 1991).

In the following sections, we cover related work, review some of the spatial problems that are present when carrying out instructional development using a virtual environment, discuss our representation system, called *tscript*, for dealing with these problems, and then cover the approach for reasoning in an intelligent tutoring system using this representation.

---

* contact mccarthy@aic.lockheed.com ph 415.354.5257 or stiles@aic.lockheed.com ph 415.354.5256

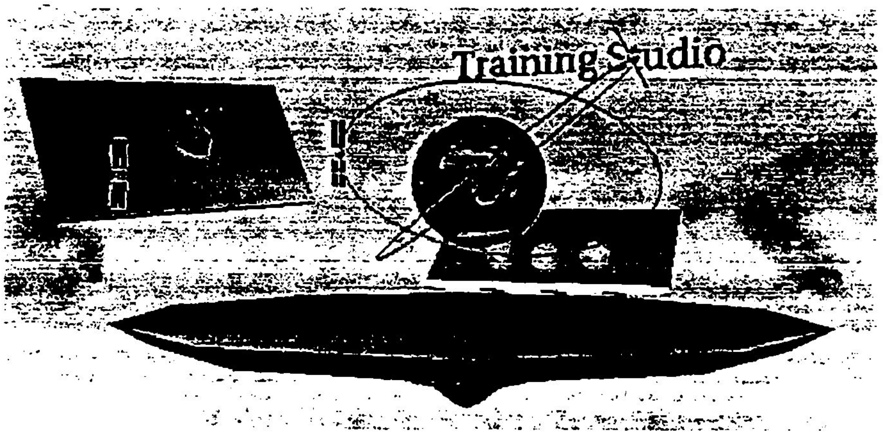


Figure 1. Training studio viewed from control room, with stage as initial reference

2.0 Background

The problem of ensuring effective understanding of three-dimensional phenomena has been addressed from a number of directions. There have been efforts to understand what attributes can enhance or degrade communication in a three-dimensional setting (Sedgewick 1991, Goldstein 1991), and others that have characterized the nature of pictorial communication and arrived at specific solutions for rescaling to achieve understanding (Ellis 1991, Eyles 1991). Then there have been attempts at automatic generation of explanatory three-dimensional scenes in response to dynamic changes in viewpoint and changing communication goals. We feel that one of the most outstanding efforts in this area is by Seligmann and Feiner, in their Intent-Based Illustration System (IBIS) (Seligmann 1991).

The IBIS system uses a collection of CLIPS-based style rules that create possible 3D illustrations and then test them for suitability. This process is handled by bodies of rules called illustrators and drafters. Illustrators are given a set of communication goals, and these goals trigger design rules that could possibly satisfy the goals. The results of the illustrator design rules, called style strategies, are passed on to the drafter. Drafters have associated with them style rules that invoke style methods. In this way, a possible 3D illustration is generated. Then the evaluation process begins. Both the design rules and style rules also have evaluation constraints associated with them, and the resulting illustration is evaluated first by the drafter's style evaluators, and then by the illustrator's design evaluators. If a given illustration strategy meets the evaluation criteria, it is then used.

Our efforts are concerned with extending this approach to address training triggered by a given locale, transitions from one locale and/or scale to another for a given training purpose, and allowing the interactive, visual specification of constraints and scenes for the delivery of instruction in a virtual environment. We do not make a distinction between illustrator rules and drafter rules, but we do adopt an generate and test approach to structure the search for an appropriate presentation.

3.0 Spatial Considerations

Many three-dimensional graphical representations, and accompanying simulations, are based on implicit assumptions about the scale they operate at, and the abstract representations that can be shown with them. These scale categories are item-based (CAD) system-based (ships, airplanes), locale-based (DIS/SIMNET battlefields) and macro-based (Planetary Systems, Orbital Representations, Galaxies, etc.). When presentation or interaction across these traditional categories is required, it is necessary to provide selected elements on several scales. For instance, training students in orbital mechanics on the distinctions between geodetic and geocentric coordinate systems may perhaps be best accomplished by providing a ground perspective for a local patch of terrain overlaid with a fine latitude and longitude grid, and drawing orientation lines from the person out to satellites above. This can be followed by launching the person's perspective smoothly out into the same orbit, at all times providing the same orientation lines from launch point to moving satellite, and then showing a transparent earth with the geocentric coordinate system underlying.

A key feature of virtual environments is the ability to change perspectives not only by manipulation of an isolated object within an environment but also by changing viewers coordinates with respect to a system. In the first case, the object's location or orientation changes, but the environment within which it exists remains constant, thus maintaining the orientation of and frame of reference for the viewer. The second condition, however, changing the coordinates of the viewer, involves a new frame of reference. Drastic changes can be disorienting, for example, watching a satellite orbit the earth at some distant point then suddenly moving to a location on earth observing the same satellite. Unless a link between the two viewpoints is established, by providing a common point of reference or some type of observable mapping, the significance of the lesson may be lost.

Changing to a completely different view can create similar orientation problems. The trainee may be required to view a location currently off-screen. If the two coordinates are distant, the two views may have nothing in common and the trainees may have difficulty placing the new location in context with respect to the previous.

Establishing a link between two seemingly disparate scenes can be accomplished before, during, or after the transition. "You are here" type maps that diagram the two locations can be displayed on either or both before and after displays. It may be useful to allow trainees to observe the transition itself; i.e., to sequence the trainee's view points over time, similar to the way a film camera pans a scene. Other effective techniques borrowed from the film industry are the concepts of zooming in (showing general scenes then moving closer to observe a particular area in detail) and zooming out (show close up views then pulling back to view the entire scene) for providing context and establishing frames of reference.

Current technology allows the representation of real life objects and processes to a great level of detail; far greater than one might need or want in an instructional sequence. Over-complexity can be detrimental to the learning conditions of the lesson since it can be difficult for learners to recognize the pertinent or critical information. This becomes especially evident when introducing complex processes. Given that trainees automatically attempt to formulate meaningful wholes from a presentation, an over-complex display can increase the probability of erroneous interpretations (Gagne, 1987). In fact, perception of critical information can be increased by limiting details (Fleming & Levie, 1978) and three-dimensional representations may not always be the most desirable for introducing concepts. Instead it may prove fruitful to introduce complex concepts or scenes as two-dimensional projections, allowing three-dimensional viewing and examination as the trainee becomes more sophisticated with respect to the domain. Some of the physics presented in "The Mechanical Universe," for example, were simplified to two-dimensional simulations to present a more understandable view of complex concepts (Blinn, 1991). There are also cases when details may be purposely blurred. For example, the earth, may be used as a link between two views. A detailed representation of the earth may appear in one scene but may be just a distant object in the current display. By providing a less detailed representation of the earth in the distance, this object becomes a reminder of earth's location without distracting the learners attention.

On the other hand, providing too little detail can have a detrimental effect since learners tend to disregard material that is too simplistic (Fleming and Levie, 1978). Additionally, the simple-to-complex characteristic of most instructional sequencing strategies suggests that levels of details should be dynamic, changing with respect to the current state of the student model. The developer, then, must have the capability of providing varying levels of detail depending upon the needs/abilities of the individual learner. Additionally, the developer must be able to specify conditions under which certain details are added or detracted.

Due to the exploratory nature of the system, a trainee's path, orientation and location just prior to a particular lesson cannot always be predicted. To provide guidance to the trainee, the developer must be able to control when certain events will take place with respect to the instructional maturity of the trainee. Strict temporal sequencing is not possible since neither the trainee's path nor time on a task is predictable. Spatial coordinates are important since certain events can only be viewed form specific locations; however, they cannot be the only triggers. The hierarchical nature of the domain content must also be considered, as well as the learning history of the trainee, and there are a variety of approaches that can be implemented to address these concerns. For example, a simulation can be activated by a spatial trigger, with granularity and details adjusted with respect to the trainee's history, or the simulation can involve a specific level of complexity and be triggered only when the learner has acquired the appropriate skills and knowledge. In either case, most instructional developers would not have the time or resources to develop separate lessons to adequately address all possible scenarios.

As discussed above, explorations and manipulations within a virtual environment involve special spatial considerations, such as orientation, level of visual detail, and granularity of simulations. To develop instructional sequences in a three-dimensional system, the developer requires a suite of tools, in addition to those relating to pedagogical and domain-related issues, that provide the following:

- Objects representing viewpoints that can be physically positioned in the virtual development. The developer should also have to ability to designate temporal attributes to these view objects to produce guided instructional sequences or "tours."
- A series of camera techniques, such as pans and zooms, to allow the developer to control or guide perspective and focus in an instructional sequence or lesson. These techniques must be represented in a form that will allow easy specification by the developer; for example, providing icons that can be attached to a view-object.
- A mapping ability to show start and end coordinates, as well as the path between. The capability to display such maps in a variety of methods (e.g., pop-up windows, overlays) is also required.
- The ability to transition to and from two-dimensional projections.
- A method of adding and subtracting detail for existing objects and processes. All objects must be scalable both individually and within a specified context.
- A method for describing domain, learner, and spatial criteria as triggers for presenting simulations and lessons. The criteria would then be placed at the appropriate coordinates in the environment and would activate specified levels of instructional sequences only when the trainee fulfilled all criteria.

All the capabilities discussed above must be supplied in forms easily recognized, manipulated, and implemented by a non-programmer. Figure 2 illustrates some of these issues using a satellite operations scenario. Eye objects are placed to guide the student's view at critical parts of the simulation. Audio can be added, and a text window provides additional information, in this case, perhaps to display telemetry corresponding to the satellite location or a two-dimensional map of the earth region covered by the satellite. Objects are exaggerated out of scale (here, the satellite is nearly the same size as earth), and realistic but unnecessary details, such as the planets between the earth and sun, have been eliminated.

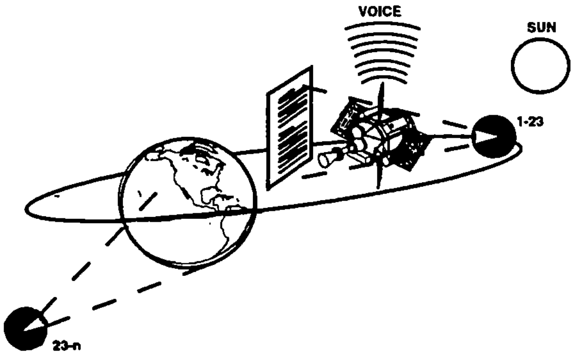


Figure 2. Sample scenario involving exaggerations of scale, two-dimensional displays, and viewpoints.

4.0 Representational System

In this section we discuss the representational system necessary for illustrating concepts in a spatially oriented, dynamic system, and how they relate to instructional development. For our purposes, an instructional unit is defined by a single learning objective. In the case of a virtual environment, the objective of an instructional unit is accomplished by presenting the student with a set of visual objects that are arranged spatially from one time step to the next in such a manner as to illustrate a concept. This scenario may be accompanied by other effects, such as sound, but that is not our emphasis.

4.1 Objects and Actors

Our framework adopts the useful object/actor breakdown for modeling a domain. An object is a model of a real or abstract entity, characterized by a state, behavior, and appearance. Objects may be simple or complex. Simple objects cannot be subdivided into sub-objects at the granularity of the model representation. A complex object is a grouping of component objects. Objects that possess intention or internal state can be considered actors. Additionally, objects can be considered either domain objects or instructional objects. Domain objects model the subject matter of the instruction. Instructional objects serve as tools for presentation and manipulation of the domain objects.

An object's state consists of attribute-value pairs which define that object's particular model of itself and the environment in which it is situated. This can be subdivided into internal and external state elements. Another useful characterization of an object's state is to partition the state into public and private elements. The publicly accessible portion of an object's state includes its appearance which is given a special distinction in the area of virtual environment modeling.

Domain objects in an instructional unit setting may serve different roles. Some objects merely provide context or cues of scale and scope. Other objects are more central to the current instructional unit. Other attributes of an object must be assigned measures of appropriateness or applicability for a given context. These attributes include rendering effects such as fixed vs. dynamic orientation shifts, transparency, material properties (surface smoothness), lighting (shadows and reflection), texturing, spatio-temporal and orientation distortion, spatial marking (highlighting) level of detail, and realism (photo-realistic to 3D and 2D stylizations). Behavioral characteristics in this category include model fidelity (high-resolution model vs. primitive lookup-tables), and temporal distortions.

At the most fundamental level an object's behavior is characterized as a collection of stimuli-response pairs. A stimulus is represented as a set of events, which are changes in the internal or external state of a given object. A response is a sequence of primitive actions; i.e., operations which cause changes in the state or appearance of an object or its external environment. In this model, when an object perceives that a specific set of events have taken place, the associated sequence of actions associated with that set of events is carried out. Environment modifying actions are realized by the sending of messages between objects either in a point to point or broadcast fashion.

An object in a virtual environment has a spatial extent or envelope. In the context of an instructional unit an object exists in 'some' state at a location or in a region (spatial envelope) for the duration of the instructional unit. More specifically, the object's location and appearance is dictated primarily by the objectives of the instructional unit in conjunction with the orientation of the viewer and objects relevant to the instructional unit.

Object Elements	Description
(scale <reference-object> <object>)	sets the scale property for a given object
(attr-relevance <object> <context>)	
(reference <geometry> <origin>)	defines a reference system using a point, geometry or object as origin
(locale <object> <viewpoint> <detail>)	defines a locale based on view, level of detail, and relevant object

4.2 Envelope Conditionals

Spatial envelopes provide a convenient means of characterizing object and inter-object behaviors. A behavior or action can be characterized by its extent in the virtual space. The triggering of a behavior is represented as the crossing of the boundary surface in a space consisting of the states of the system and space and time. For many practical situations the triggering of a behavior is simply a matter of crossing the threshold of a particular spatial or temporal envelope when a particular object state is present.

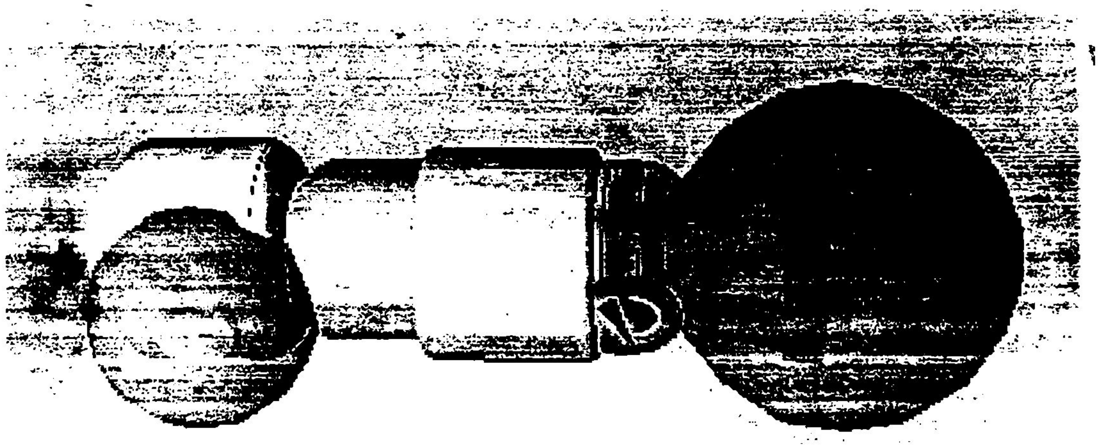


Figure 3. Spatial envelopes associated with rocket engine nozzle and inlet

An example would be the envelopment of an object in a nested spherical enveloping-regions with graduated sizes, with a different appearance rendering behavior attached to each sphere. The viewer' visual focus (eye) crossing an envelope relevant to an instructional unit would cause the rendering behavior associated with that envelope to be used to render the object. Moving closer to the object could cause renditions of increasing level of detail to be used for viewing the object. This is often done in visual simulations.

Envelope conditionals can be modified to be applicable only at a given scale by adding another conjunctive conditional wherever they appear that explicitly references scale of operation. Alternatively, they can be made applicable across scales by utilizing the fact that each envelope associated with an object is attached as a child in the transformation hierarchy, and will be appropriately scaled and translated whenever its associated object is.

Envelope Elements	Description
(envSphere <object> <radius> <opt-scale>)	specifies spherical envelope around and object or point
(envRect <object> <x-extent> <y-extent> <z-extent> <opt-scale>)	specifies bounding box envelope
(envGeometry <object> <scale-factor>)	specifies more detailed envelope, using a new scaling factor applied to the geometry

4.3 Viewing

Viepoints onto a given scene are used during the specification of transitions, and by themselves in the context of an instructional unit. These viewing elements can be used as conditionals that trigger actions associated with an instructional unit, including level of simulation granularity, relevant explanation actions, or display actions. In display actions, the viewing elements of *tscript* are used on the left-hand-side of generative rules serve to constrain the kinds of display actions or transitions that are generated. When they appear on the left-hand-side of evaluation rules, they are used as tests, determining whether or not the display objective has been met. The elements of our representation that describe constraints on views follow.

Viewing Constraints	Description
(viewPos <viewer> <object> x y z)	specifies view position relative to a given object's center, commonly known as tethering
(viewAnchor <viewer> <object>)	specifies orientation of viewer relative to another object
(viewGlobalPos <viewer> x y z)	specifies position of viewer in global coordinates always overrides any conflicting view
(viewReference <viewer> <object> <reference>)	specifies that reference system should be present in scene
(viewNormal <viewer> <object>)	view object normal to surface
(viewVisible <viewer> <object>)	ensure object is visible in any resulting scene
(viewEntire <viewer> <object>)	ensure all of an object's bounding box is visible in any chosen scene
(viewContain <viewer> <object1> <object2>)	make sure view contains both objects in relation to each other, independent of scale

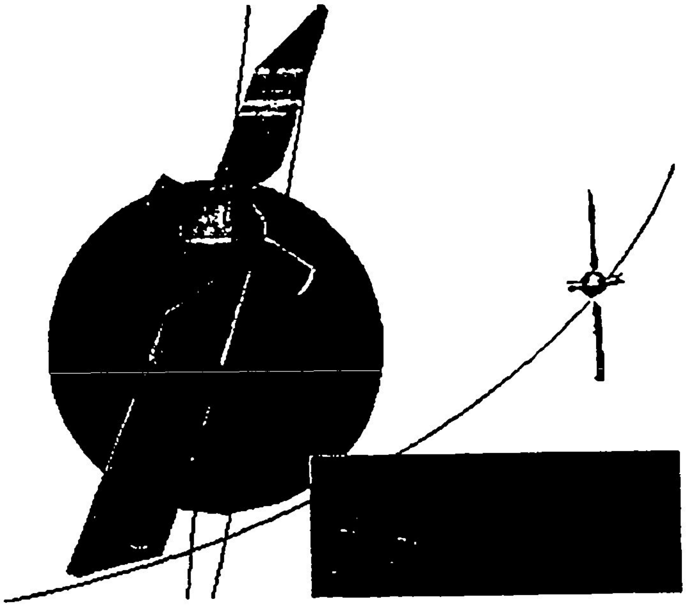
4.4 Transitions

The envelopes of objects relevant to an instructional-unit may include vast areas of transition (dead zones) between regions of interaction. This motivates the need for a representation of these transitions allowing manipulation and reasoning about transitions. Methods for handling spatial or temporal transitions include:

- path-following - travel along a given path to impart a sense of continuity.
- inset transition - travel out to destination showing inset of original location (see Figure 4).
- teleporting - abrupt discontinuous movement from one place or time to another.
- warping - Constitutes a transition where compression or expansion of time or space when traveling between two locales is used. An excellent example of this is logarithmic travel where an observer travels the same relative percentage toward a given object with each time step, initially traveling very fast, and gradually slowing down as the target locale is reached (Mack 1990).
- shift exaggeration - A modification of teleporting where cinematic style transitions such as wipes, fades, zooms, pans are used.

Shift exaggerations do not conform to physical laws, but have been shown to be quite effective cues to the observer. These techniques can be used as a subtle reinforcement to highlight a transition when applied in combination with the other more physically consistent transition methods.

Figure 4. Inset transition from locale scale to macro-scale in orbit



Transition Element	Description
(tranInset <view1> <view2>)	specifies transition using inset of original vie prior to start of transition
(tranWarp <view1> <view2> <percentage>)	specifies a warp transition type, with percentag being the amount of remaining distance covered a each step of the transition
(tranTeleport <view1> <view2> <overlay>)	specifies a teleport transition type, with optiona overlay
(tranPath <view1> <view2> <path-list>)	specifies a path transition over a defined path
(tranRoadMap <view1> <view2> <reference>)	specifies transition with roadmap associated wit different scale inserted between
(tranSurface <view1> <view2> <object>)	specifies transition across surface
(tranDirect <view1> <view2>)	specifies shortest line transition

4.5 Level of Detail

Displaying geometric objects at different levels of detail based on the distance from a viewer is a practical approach to supporting interactive rates in virtual environments. When visible objects are distant, they can be replaced with less polygonally detailed models, and thus the frame rate for rendering these objects can be increased. This principle can be useful for instructional purposes as well. As Fleming & Levie (1978) note, the learner does not always know what is the most critical information in a complex display. To minimize the discrepancy between the nominal stimulus (what the designer meant by the display) and the effective stimulus (what learner attends to), the essential part must be salient, dominant, and noticeable (Gagne, 1987). There are cases when instructional material must be presented in less detail, and then further detail is introduced when it becomes relevant.

The easiest qualification on detail is a simple number, with 1 being the highest level of detail available in the system, and increasing numbers indicating lessening detail. This is easily supported under tree-based graphics systems such as IRIS Inventor (Strauss 1992) by testing distance and then selectively displaying nodes by switching between them, or IRIS Performer (McLendon 1992) by using built-in level-of-detail testing. Qualitative constraints on detail are important as well. For example, in the case where a student is being familiarized with an engine assembly, detail including individual bolts is not necessary. But as instruction progresses, and stripping down the engine is the focus of the instructional unit, this level of detail becomes appropriate.

Detail Element	Description
(detail <object> <detail-level>)	specifies detail using numerical detail level, ca override default checks based on distance fro object center
(detail <object> <detail-object>)	specifies detail level that has a given sub-objec recognizable and included in scene

4.6 Visual Specification

We are working on the visual specification of *tscript* elements. Transition types and spatial envelopes will be represented by iconic geometric structures. In the case of envelopes, a simple system of attaching an icon to a given domain object and then stretching it out to the appropriate size will be supported. For *tscript* elements that require object arguments, direct selection of each object is appropriate, initially using a mouse, and eventually using a glove interface. Transitions pose a larger problem because they can involve radically different settings. This can be handled by reducing locales themselves to small icons in the training studio control room.

5.0 Instructional Approach

This section relates the *tscript* representational system to the characteristic intelligent tutoring system (ITS) framework. Most intelligent tutoring system include an expert module, a student module, an instructional module, an interface, and a simulation (Regian 1991). The *tscript* reasoning system should be considered part of the instructional module. One might at first assume *tscript* is a natural part of the interface system, but *tscript* allows specification of, and reasoning across, a spatial domain. For virtual environments, the domain of instruction is mapped onto a spatial representation. Pedagogy expressed in *tscript,* and concerned with the appropriate way to present spatial and geometrical information, is essential.

5.1 Limiting Evaluation

Through the application spatial envelopes, the search for appropriate presentation of objects in the training environment can be limited to those objects which are relevant to the student at a given locale. Of course, the first and primary criteria for limiting evaluation is the student model (see Figure 5). At the point where a set of actions are possible using the student model, they also are matched conjunctively against any applicable spatial envelope conditions, using viewer coordinates. At this point, a smaller set of instructional strategies are generated. These are then evaluated against viewing and detail criteria to result in a decision on the appropriate instructional diaplay.

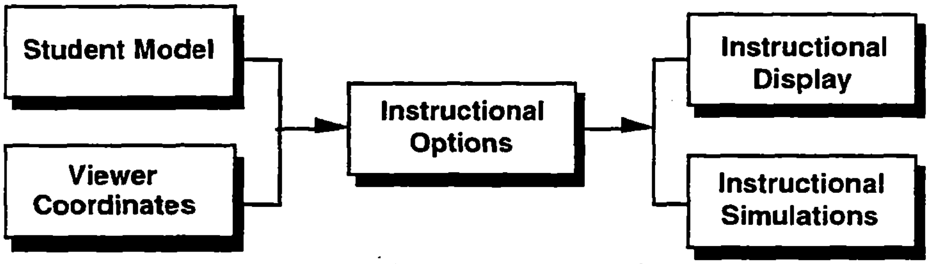


Figure 5. Viewer coordinates and normal student model info determine instructional options

5.2 Prioritizing Spatial Responses

Under the dynamic conditions supported by a domain simulation, it is possible for viewing constraints stated by the instructional developer to be in conflict. To resolve conflicts, we associate a dynamic measure of importance with each object. This measure ranks the objects in priority for computing resources and the attention of the student. The priority can't be a simple quantity because it must be able to express the priority of an object along multiple dimensions. These dimensions include: behavioral fidelity, rendering fidelity, and relevance to the instructional unit. This priority will determine which of several competing instructional actions will be used.

6.0 Summary and Future Work

Virtual environments couple visual, spatial representations with one or more simulated aspects of reality. When they are used in support of training, there must be support for visual presentation of instructional concepts, which may not always be subordinate to realistic display and interaction. We have highlighted a set of visual effects that are necessary for effective instruction in a virtual environment, and presented a representational system for defining and reasoning about these visual effects in the context of a dynamic, simulated environment.

Our future work will focus on visually specifying the elements of *tscript*, building a comprehensive set of heuristic 3D display rules, and on refining the interaction between the *tscript* presentation rules and the student module.

References

Blinn, J.F. (1991). The making of The Mechanical Universe. In S. Ellis, M. Kaiser & A. Grunwald (Eds.) *Pictorial Communication in Virtual and Real Environments* (pp. 138-155). New York, NY: Taylor & Francis, Ltd.

Ellis, S. R. (1991). Pictorial communication: pictures and the synthetic universe. *Pictorial Communication in Virtual and Real Environments* (pp. 22-40). New York, NY: Taylor & Francis, Ltd.

Eyles, D. E. (1991). A computer graphics system for visualizing spacecraft in orbit. In S. Ellis, M. Kaiser & A. Grunwald (Eds.) *Pictorial Communication in Virtual and Real Environments* (pp. 196-206). New York, NY: Taylor & Francis, Ltd.

Fleming, M., & Levie, W. H. (1978). *Instructional message design.* Englewood Cliffs, NJ: Educational Technology Publications.

Gagne, R. M. (1987), *Instructional Technology: Foundations.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Gagne, R. M., & Driscoll, M. F. (1988), *The essentials of learning for instruction, 3rd edition.* Fort Worth, TX: Holt, Rinehart and Winston.

Grant, F., McCarthy, L., Pontecorvo, M., Stiles, R., (Nov. 1991). Training in virtual environments. *Conference on Intelligent Computer-Aided Training,* Houston, TX.

Mackinlay, J. D., Card, S. D., Robertson, G. G. (1990) Rapid controlled movement through a virtual 3D workspace. In *SIGGRAPH Conference Proceedings,* (pp. 171-176). Dallas TX: ACM Press.

McLendon, P., Phillips, C., Helman, J. et. al. (1992), *IRIS performer programming guide,* Mountain View CA: Silicon Graphics, Inc.

Regian, J. W., (1991). Representing and teaching high performance tasks within intelligent tutoring systems. In H. Burns, J. Parlett & C. Redfield (Eds.) *Intelligent Tutoring Systems: Evolutions in Design* (pp. 225-241). Hillsdale N. J.: Lawrence Earlbaum & Assoc, Inc.

Seligmann, D. D., Feiner, S., (1991), Automated generation of intent-based 3D illustrations. In *SIGGRAPH Conference Proceedings,* (pp. 123-131). Las Vegas NV: ACM Press.

Stiles, R., Pontecorvo, M., (Sept. 1992) Lingua graphica: a visual language for virtual environments. *Proceedings of the IEEE International Workshop on Visual Languages.* (pp. 225-228). Seattle WA: IEEE Press.

Strauss, P. S., Carey, R. (July 1992), An object-oriented 3D graphics toolkit. In *SIGGRAPH Conference Proceedings,* (pp. 341-350). Chicago, IL: ACM Press.