Semantic Instrumentation of Virtual Environments for Training

Christian Greuel, Grit Denker, Karen Myers
SRI International
Menlo Park, CA
christian.greuel@sri.com, grit.denker@sri.com, karen.myers@sri.com

ABSTRACT

Interactive virtual environments are useful tools for hands-on learning or rehearsing of procedural tasks. However, task training applications typically provide a constrained course of action for the learner, forcing them down a single specific solution path. We discuss an approach in which the virtual environment is semantically instrumented in order to allow for the tracking of and reasoning about open-ended learner activity therein. Our approach leverages ontology-based knowledge representation which allows for a structured and meaningful description of virtual objects and of the learner actions that may be performed upon them. This is facilitated by the association of specific ontological classes with geometric components of the objects which populate a training exercise. These classes, together with their attributes, relationships and rules, characterize the environment and user actions in a readily understandable manner. As a result, a training system is able to observe the learner activity, render an assessment of that activity, and provide meaningful feedback to the learner. We also present an authoring tool which allows content developers to semantically annotate three-dimensional models for such an environment.

ABOUT THE AUTHORS

Christian Greuel is a Senior Research Artist in the Computer Science Laboratory at SRI International, with a focus on visual computing. His areas of applied research include design of tools, techniques, and processes for representation of and interaction with virtual environments; web-based training applications; geographic visualization; geometric modeling; multi-modal data representation; and immersive audio. He earned an MFA in Performing Arts Design and Technology from the California Institute of the Arts.

Grit Denker is a Program Director in the Computer Science Laboratory at SRI International. Her expertise is in automated reasoning applications using semantic technologies, ontologies, and rules; cognitive and policy-based systems; specification and verification of security, network and privacy policies in distributed systems; and novel human-machine interface technology. She has a Ph.D. in Computer Science from the Technical University of Braunschweig, Germany.

Karen Myers is a Principal Scientist in the Artificial Intelligence Center at SRI International, where she conducts basic and applied research in intelligent systems. Her specific areas of expertise include autonomy, intelligent assistants, mixed-initiative planning and control, and programming by demonstration. In addition to being widely published, she has overseen several successful transitions of her research into use both by the U.S. Government and within commercial settings. She has a Ph.D. in Computer Science from Stanford University.

Semantic Instrumentation of Virtual Environments for Training

Christian Greuel, Grit Denker, Karen Myers
SRI International
Menlo Park, CA
christian.greuel@sri.com, grit.denker@sri.com, karen.myers@sri.com

INTRODUCTION

Virtual environments (VEs) provide an appealing vehicle for acquiring procedural skills, particularly in domains where real-world training incurs significant time, expense, or risk. Taking full advantage of VEs, however, requires mechanisms to assess learner performance, such as direct observation by an instructor. Mechanisms that enable automated assessment can both reduce the cost of using VEs for training and open the door to self-directed learning systems, in which users can acquire procedural skills at their own pace and on their own time.

SRI International (SRI) developed a prototype framework, Semantically Enabled Automated Assessment in Virtual Environments (SAVE) (Greuel, Myers, Denker, and Gervasio, 2016), capable of observing a learner operating within a semantically instrumented VE. The framework was designed to automate the assessment of the learner's performance and provide helpful feedback to improve their skills. In contrast to intelligent tutoring systems that address "algorithmic" skills with a single or small number of acceptable responses, the SAVE framework addresses more open-ended procedural skills that can have a range of acceptable solutions.

For the automated assessment capability to understand learner activity and provide meaningful feedback, the VE must include a semantic characterization of both the environment and user activities. For example, valuable assessments can be made in a home construction exercise if the system knows which drill bit is required to bore through masonry, the minimum distance that a heating exhaust vent may be place from a residential window, or the correct order for installment of plumbing, electrical, insulation, and drywall.

VEs today are typically closed systems that do not provide generalized instrumentation beyond what is needed for internal gameplay. The logic is usually application-specific and hardcoded in the software. New tools are needed to allow content developers to semantically annotate virtual objects and thus enable automated scene understanding and assessment of learner performance.

STATE OF THE ART

The semantic instrumentation of VEs goes back at least to Winograd (1971), where it was used to allow a user to convey natural language text instruction to a (non-rendered) virtual robot. Semantic tagging of three-dimensional (3D) models was explored years later within the cultural heritage community to improve curation of static artifacts such as statues and monuments (Rodriguez-Echavarria, Morris, and Arnold, 2009). SAVE, by comparison, leverages semantic annotation to characterize user interaction with dynamic 3D objects.

In recent years, various systems have explored the use of semantics in VE-based training. Most of these assume predefined semantic models embedded in the VE, although there are exceptions. (Kessing, Tutenel, and Bidarra, 2012) describes a framework in which game objects are semantically classified using authoring tools that assign tags from database libraries. However this approach is applied to simple two-dimensional (2D) bitmap sprites, whereas SAVE works with complex, articulated 3D objects.

(Maderer, Gütl, and Al-Smadi, 2013) describes a training architecture that uses an external repository to hold information about virtual 3D objects. Tools allow for authors to associate metadata with a place mark or simple object, such as a spatial zone for proximity testing. However, embedded functionality, such as for describing a user interaction with an object, still requires a software developer to extend the application code. SAVE authoring tools allow for granular annotation of objects and their subcomponents, to enable tracking of complex user interactions. Further, SAVE tools also allow for the visual definition of object behavior resulting from specific user interactions.

APPROACH

The process of semantically instrumenting the environment begins with traditional 3D models, which are enhanced with semantic annotations. An authoring tool for intuitive annotation of the 3D models is also presented.

Traditional 3D Models and Scene Graphs

The most recognizable element of a typical VE is the surface level representation known as the *3D model*. This is a geometric mesh that defines the basic spatial structure and visual qualities of one or more virtual objects. This mesh however is little more than an empty shell comprised of faceted surfaces, onto which 2D raster images (known as *texture maps*) are projected to provide the illusion of some surface material, such as stone, metal, or wood grain when rendered (Figure 1). Three-dimensional models are created by a 3D artist, or *modeler*, using third-party *digital content creation* (DCC) tools[1]. A number of online repositories provide ready-made 3D models for download. However, careful attention must be paid to the wide ranging quality, fidelity, and licensing agreements of such products as they relate to the intended use. Regardless of whether they are acquired or custom built, 3D models may still need to be edited for segmentation so that they can be articulated and annotated at the proper granularity.

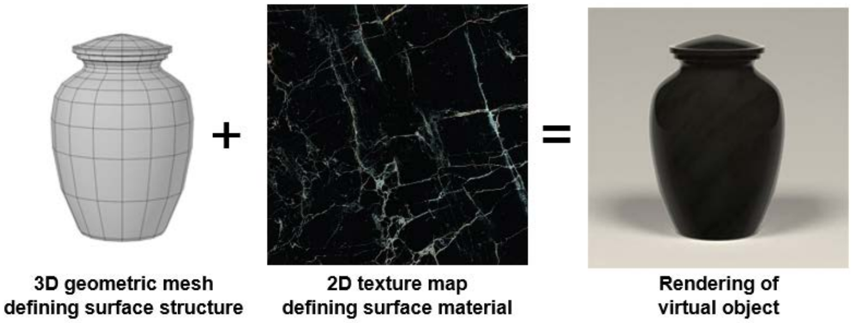


Figure 1. A 3D model is a geometric mesh that visually defines an object surface and its material appearance.

Any number of 3D models are collected and organized in the form of a scene graph, which is the core data structure used for the logical representation of graphical scenes. The scene graph is a hierarchy of mesh nodes, containing basic spatial information for each object (Figure 2). Nodes can be freely named by the modeler or according to some agreed convention. Hierarchical organization is often arbitrary, but it is commonly used to optimize rendering performance by grouping objects by type, location, or assembly. However, the traditional scene graph normally does not carry higher-level characterization of object identity, purpose, capabilities, or sub-surface physical qualities. The scene graph content, when rendered, allows for the illusion that objects—such as a collection of tools on a workbench—occupy a space, but it provides no concept of object properties, whether informative (e.g. manufacturer name, part number, or material composition) or actionable (e.g. operable, removable, or combustible). In rare cases where metadata is leveraged, this is usually embedded with the node as a plain text string using an application-specific syntax. A generalized approach is needed to enable the attachment of semantic meaning to virtual objects.

Semantically Enhanced 3D Models

Semantic Annotation: Beyond the visual representation of an object, a semantically enhanced 3D model can also provide its attributes such as its identification, functionality, material characteristics, or just about any property deemed relevant to an exercise. Further, it can describe an object's dynamic properties relative to user actions (e.g. a switch is engaged) and in relationship with other objects (e.g. an item is held in a container). The attribute collection is reusable in other exercises and can be carried forward, becoming more fully defined over time, leveraging previous work by other authors and benefitting future authors.

[1] Common DCC tool examples include Autodesk Maya, Autodesk 3ds Max, Presagis Creator, and Blender for the creation of 3D models, as well as Adobe Photoshop and GIMP for texture map creation.

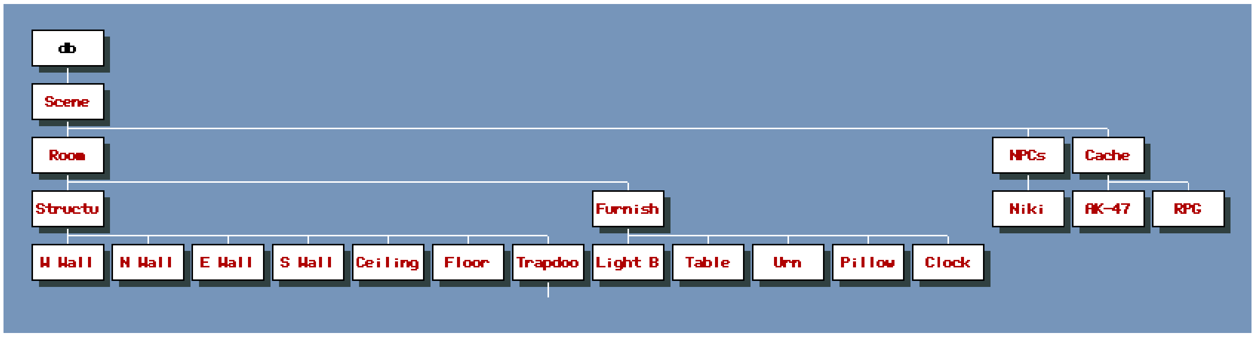


Figure 2. A scene graph is a hierarchical grouping of 3D models, containing basic spatial and grouping information. However, scene graphs typically do not contain any semantic characterization.

These meaningful properties, stored in the form of an ontology, are used by SAVE's automated assessment module to enable comparisons between a "gold standard" exercise solution and a detailed log of the learner's activity. For example, if a student taking a course in construction chooses an insulation-enhanced plastic sheeting for the roof of a house, the sheeting properties could enable the assessment to validate the satisfaction of a weatherization requirement in the exercise solution. If a budget were associated with this construction exercise, the assessment could infer whether the increased cost of an insulation-enhanced plastic sheeting would violate a budget constraint.

To associate semantic concepts with them, 3D models are annotated by reference to classes defined in an ontology (Figure 3) stored within a knowledge base (KB). This is purposefully kept separate from the 3D models. This not only allows the models to be modified or exchanged as needed, but it also allows the ontology to be updated or expanded without requiring the models to be edited. This separation provides for greater flexibility and reusability than if such annotation were embedded directly in the 3D model file.

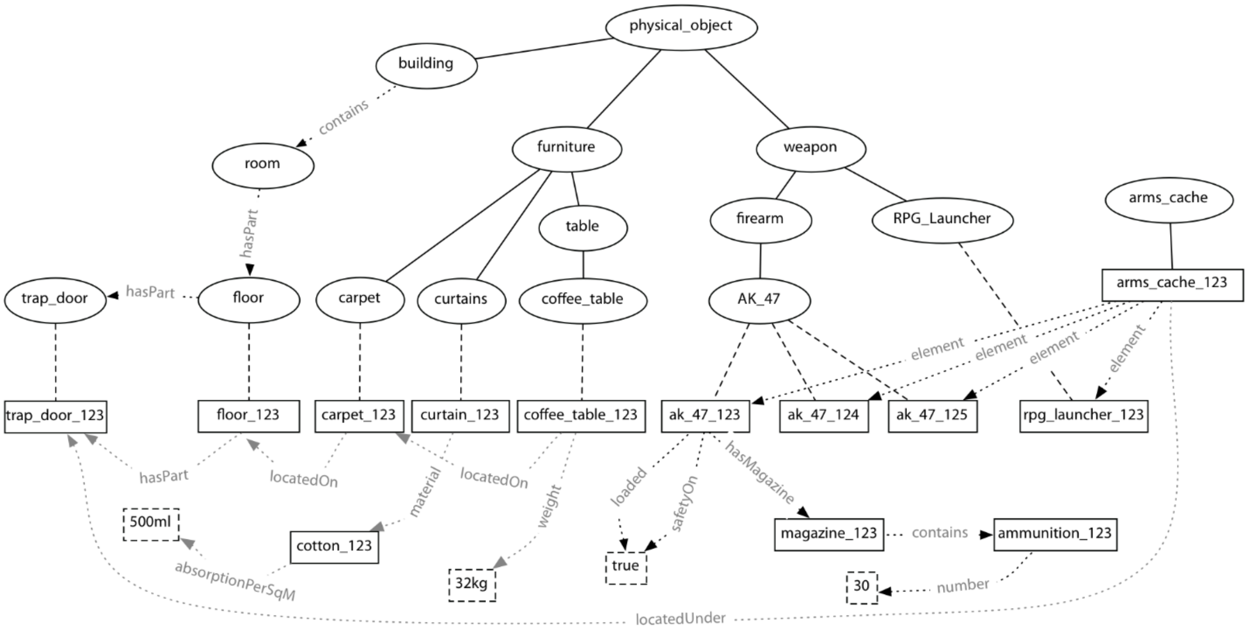


Figure 3. 3D models (solid rectangles) are annotated by reference to ontological classes (ovals). Semantically enhanced They can be further defined by property attributes (dashed rectangles) and relationships (arrows).

A *semantic 3D* (S3D) file, formatted in Extensible Markup Language (XML), documents the mapping of the 3D model geometry nodes to appropriate ontological classes (Figure 4). The "flora_base" element identifies the location of an external ontology in which classes are defined. The "asset" element refers to a specific 3D model file (e.g. an AK-47 rifle) and annotates it with one of the classes. Assets may contain one or more "object" elements. Each represents an individual component that is to be semantically addressable as well (e.g. a magazine, a selector lever, a charging handle, or a trigger) and maps a class reference to the geometric node.

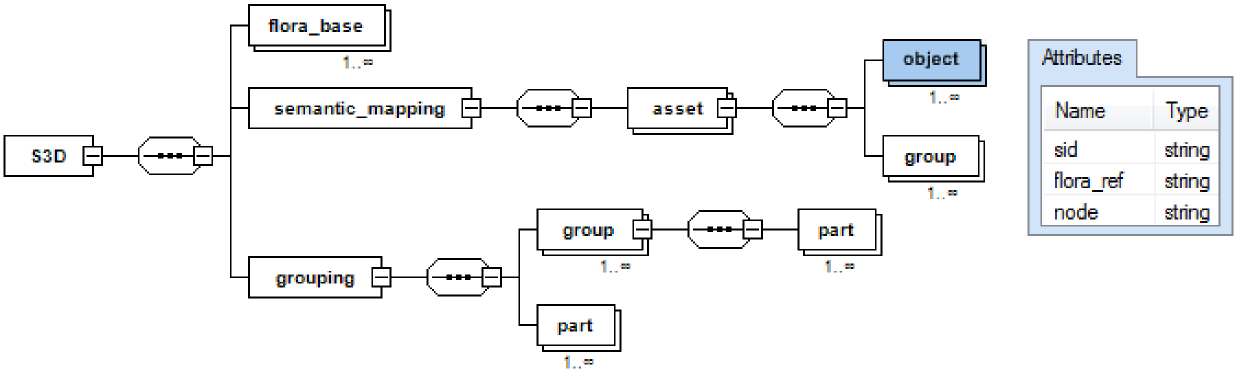


Figure 4. XML Schema for the S3D file, which documents the mapping of 3D model geometry nodes to appropriate ontological classes. Attributes of the "object" element are shown.

The "grouping" element is used to allow for dynamic control of the objects as defined collections, such as component subassemblies. This grouping data is independent of the 3D model's static scene graph hierarchy, described earlier, although the latter can certainly be used as a starting point for the former. Each model component is identified here either as a part contained within a specific group or as a separate part. The semantic mapping of these assembled groups then occurs under the "asset" element, in the same manner as an individual object would be.

Ontologies: An ontology is a structured form of knowledge representation that describes classes (e.g., types of objects), their generic properties (e.g., material, enumeration), and relationships with other classes (e.g., inheritance, composition). Specific properties values can be either defined or derived when an object is instantiated. For example, a window object may be queried to learn which materials it is constructed of (e.g., aluminum and double-glazed glass), its size (e.g., width and height), and how it operates (e.g., double hung or vertical lift). Relationships can be defined in terms of other objects (e.g., that a window is installed on a given wall).

For representation of user activity, classes are defined for individual actions. This allows for a semantic assessment of the steps performed by the learner. For example, to remove an old car battery one must first check to ensure the engine is off, then disconnect the negative cable, followed by the positive cable, and finally remove the battery itself. Exercise solutions in SAVE are formulated as a series of such actions. That is, the sequence(s) of actions performed on objects in the VE, together with constraints on those actions (e.g., the negative cable must be disconnected before the positive cable in order to avoid a short circuit), together define how the task is correctly performed. Since user activity is tracked, a domain expert can author such exercise solutions simply by demonstration the procedure in this same VE. The authoring process is described further in (Greuel et al., 2016).

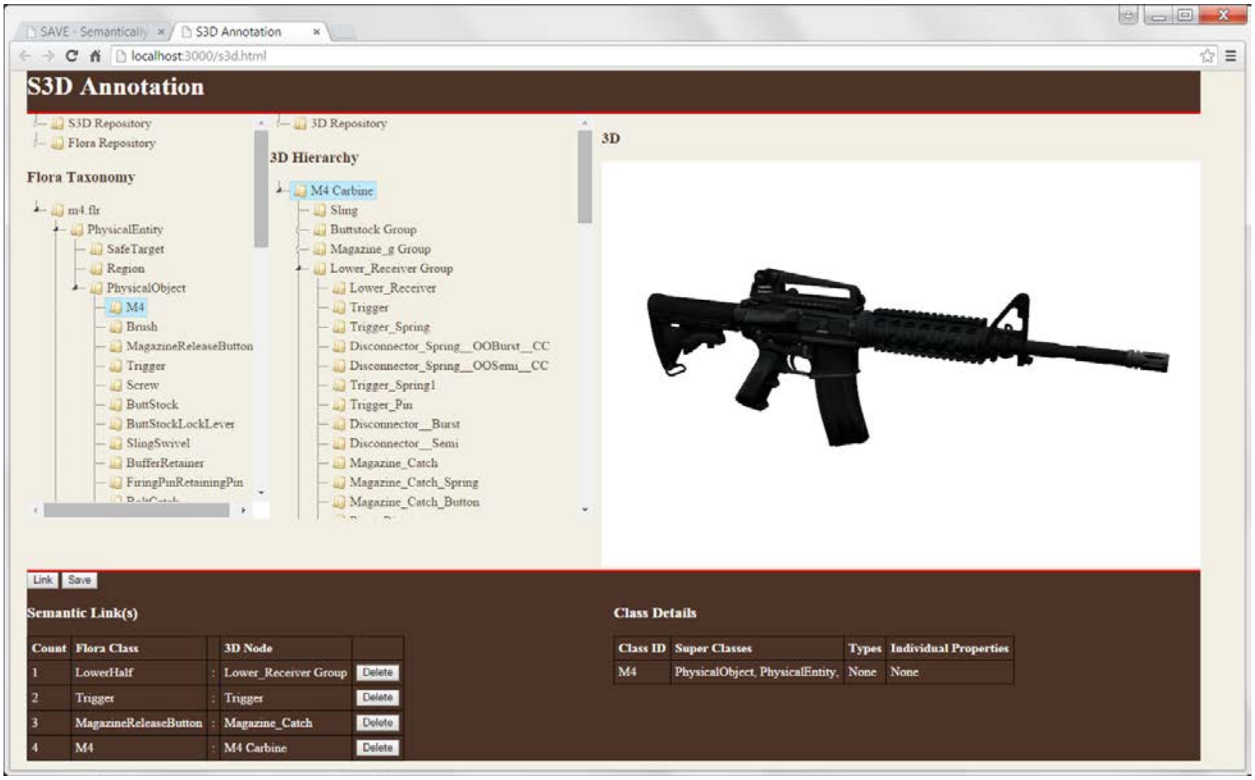
Rules: Ontologies can describe the semantic structure and properties of objects. Alone, however, they are not sufficient for describing computations (e.g. functions) or dynamic behavior (e.g. how object properties change through interaction with the objects). Rules are used to describe these more complex constructs. Rules can be used to define derived properties that are either quantitative (e.g., area, distance, weight) or qualitative (e.g., tall, nearby, heavy). Rules can also be used to describe how the learner's interaction with an object changes its properties (e.g., disengaging a safety switch will enable a trigger to be operated). Alternatively, external engines or procedural attachments could be used to compute more complex forms of derived properties, such as the force with which a falling object would collide with a surface, or whether something would be visible through a window.

The *Flora-2* rule-based system[2] is used for the underlying knowledge representation and reasoning in SAVE. This is a highly expressive language that meets the representational needs for definition of both classes and rules. *Sunflower Foundation*[3] is a software library, developed by SRI, which provides many features that are essential to building

[2] http://flora.sourceforge.net
[3] http://sunflower.csl.sri.com

applications based on Flora ontologies and rules. A detailed description of how these are developed and implemented in the Sunflower environment for use within the SAVE system is presented in (Elenius, Denker, and Kim, 2016).

Authoring Tool



Semantic Link(s)

Count	Flora Class		3D Node	
1	LowerHalf	:	Lower_Receiver Group	Delete
2	Trigger	:	Trigger	Delete
3	MagazineReleaseButton	:	Magazine_Catch	Delete
4	M4	:	M4 Carbine	Delete

Class Details

Class ID	Super Classes	Types	Individual Properties
M4	PhysicalObject, PhysicalEntity,	None	None

Figure 5. The S3D Annotation tool provides a user interface to simplify the semantic markup of 3D models.

The *S3D Annotation tool* (Figure 5) was developed to simplify the semantic markup of 3D models. This tool loads and displays an existing 3D model in a graphical user interface and allows a content developer to annotate the model and its subcomponents with the semantic information stored in the KB without the need to write software code. The S3D tool leverages metaphors and interface elements, such as WYSIWYG (what you see is what you get) rendering, 3D scene navigation, and hierarchy editing tools, which are already familiar to modelers. To these capabilities was added that to create semantic links between ontological classes and the 3D model components.

For our initial effort, a custom ontology was engineered to represent the domain of small arms maintenance for the M4 series rifle carbine (Elenius et al., 2016), providing exercise appropriate classes that a content developer was able to use for the annotation of a 3D model representing the M4. Once a 3D model is annotated, its properties become available in any SAVE exercise in which it is included. Semantic annotation of 3D models can therefore be a one-time effort. If new classes are later developed in the ontology and an author wanted to take advantage of them, he would only need to add new semantic annotations to the appropriate components of the 3D model. However, if existing classes are simply being extended with new properties, then changes are not even needed to the annotation, since the semantic link has already been made.

An anticipated update to the S3D tool will incorporate *behavior authoring* (Figure 6) to enable a content developer to visually define end-states of a 3D model component, representing the result of any action that may be acted upon it. For example, if the author has a 3D model of an automotive gear selector and has an automotive ontology enumerating actions to select modes for “Park”, “Reverse”, “Neutral”, “Drive”, and “Low”, then desired gear selector geometry location for each of these actions can be specified. As a result, when a learner chooses within an exercise to place the gear into a certain mode, the selector can be visually moved into the corresponding position.

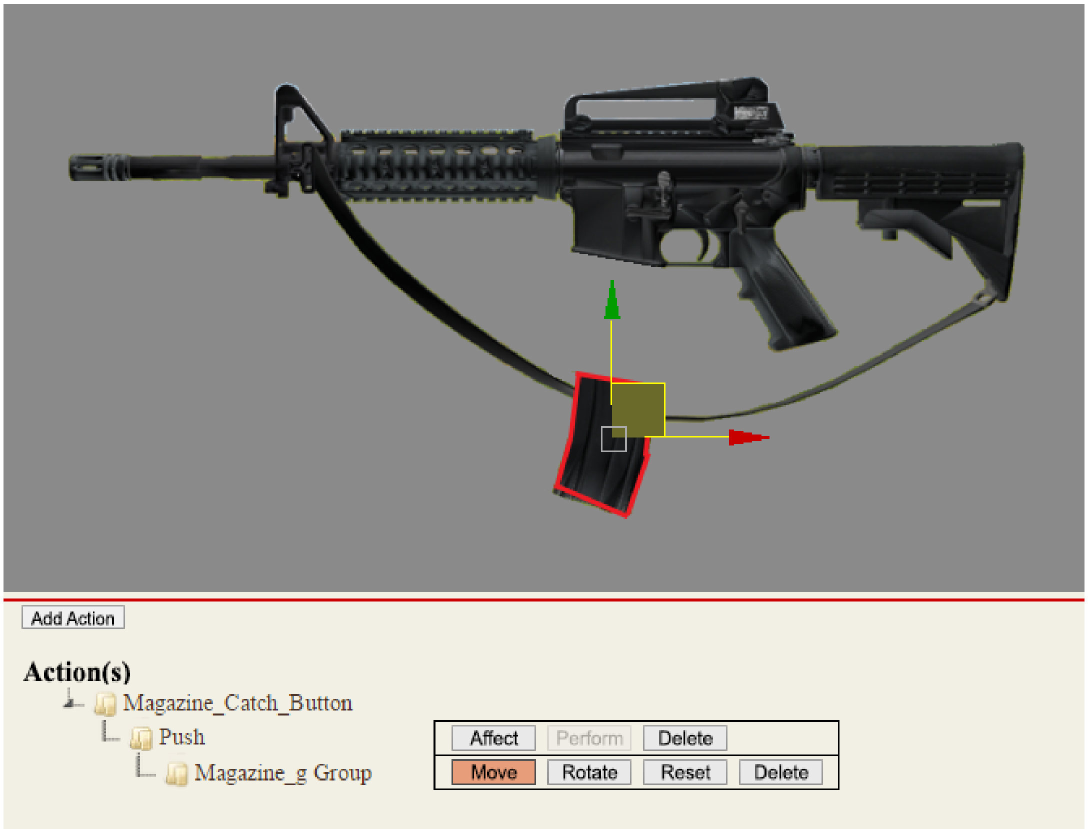


Figure 6. Behavior authoring enables a content developer to visually define end-states of a 3D component, representing the result of an action that may be acted upon it. Here, it is specified that a 'push' action applied to the magazine catch button will cause the magazine to move downward.

BENEFITS VS. LIMITATIONS

The use of VEs for procedural training is an active area of research, with a breadth of topics that should continue to be explored to fully realize the promise of intelligent VE-based training. The work presented here is part of a novel combination of research thrusts that together resulted in the SAVE framework that provides an end-to-end capability, from exercise authoring to learner experience and assessment. It allows for the rich semantic annotations of a VE, a feature not found in traditional scene graphs, enabling the automated assessment of open-ended procedural task performance.

The SAVE assessment capability was developed specifically to support the hands-on practice and rehearsal aspects of learning that are critical to mastering procedural skills. It was not intended to address guided performance for task familiarization, which is typically better-served by constrained content in a limited branching navigation structure. However, semantically annotated content could certainly be utilized in the latter context, e.g. as a means for the learner to access additional information or learning resources related to components of interest.

SAVE itself has no explicit user model and does not represent or track the learner's knowledge about the training domain. The framework could however be integrated with complementary instructional technologies by implementing e-learning communication protocols such as the Experience API[4] (xAPI).

[4] http://experienceapi.com

FUTURE WORK

The framework as developed to date does not immediately support robust virtual reality or video gaming applications. However, integration with game platforms, physics engines, alternate user input modalities, and/or stereoscopic display devices, would enable more immersive user experiences to leverage this framework.

Additional development could increase the functionality and usability of the prototype. One significant area under consideration is the enrichment of our core representations to include explicit models of state. Doing so would enable the capture of training requirements related to changing conditions in the VE (i.e., what was previously accomplished) rather than to actions alone (i.e., what was just executed), as in the initial prototype.

Provenance data, identifying which author annotated which objects, and when they did so, could be added as meta-data to provide a basis for different authors to understand how the semantic properties of objects had been previously modified by others. Simple consistency checks could be made, for example, to ensure that the addition of an annotation does not conflict with current annotations for an object.

CONCLUSION

This paper presented the technique of semantically instrumenting virtual environments that is utilized by the SAVE framework in support of training open-ended procedural tasks. The approach treats semantic specification as a first-class authoring activity and provides an interactive tool to attribute semantics to elements that comprise the virtual training environment rather than requiring system developers to hard-code this information in advance. Semantically enabled environments allow for the tracking of user activity at a meaningful level, the authoring of exercise solutions by means of demonstration, and the automated assessment of learners undertaking those exercises.

ACKNOWLEDGEMENTS

This paper is based upon work supported by the United States Government under Contract No. W911QY-14-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the Government.

The authors thank Daniel Elenius, Melinda Gervasio, Chris Jones, Scott Krehbiel, John Pywtorak, Rukman Senanayake, and Susanne Riehemann for their contributions to the work described in this paper.

REFERENCES

Elenius, D., Denker, G., & Kim, M. (2016). Semantically enhanced virtual learning environments using Sunflower. In *Proc. of 10th International Conference on Metadata and Semantics Research*. Göttingen, Germany.

Greuel, C., Myers, K., Denker, G., & Gervasio, M. (2016). Assessment and content authoring in semantically enabled virtual environments. In *Proc. of Interservice/Industry Training, Simulation and Education Conference*. Orlando, FL.

Kessing, J., Tutenel, T., & Bidarra, R. (2012). Designing semantic game worlds. In *Proc. of the 3rd Workshop on Procedural Content Generation in Games*.

Maderer, J., Gütl, C., & Al-Smadi, M. (2013). Formative assessment in immersive environments: A semantic approach to automated evaluation of user behavior in Open Wonderland. In *Proc. of the Immersive Education (iED) Summit*. Boston, MA.

Rodriguez-Echavarria, K., Morris, D. & Arnold, D. (2009). Web based presentation of semantically tagged 3D content for public sculptures and monuments in the UK. In *Proc. of 14th International Conference on 3D Web Technology*. Darmstadt, Germany.

Winograd, T. (1971) Procedures for a representation for data in a computer program for understanding natural language. Massachusetts Institute of Technology.