SRI International

23 January 2015

SEMANTICALLY ENABLED AUTOMATED ASSESSMENT IN VIRTUAL ENVIRONMENTS (SAVE) – VERSION 1.1

USER GUIDE

Project Document No: SAVE 15-002

333 Ravenswood Avenue • Menlo Park, CA 94025-3493 • 650.859.2000 • www.sri.com

CONTENTS

FIGURES

1. INTRODUCTION

1.1 ABOUT SAVE

SAVE (Semantically Enabled Automated Assessment in Virtual Environments) is a framework that can observe a learner operating within an instrumented virtual environment (VE), assess learner performance, and provide helpful feedback to improve learner skills. SAVE includes tools that support creators of online training systems in specifying instructional content, leveraging the visual nature of the VE to provide an intuitive 3D authoring environment.

Understanding what the learner is doing is the key to enabling automated assessment. Our approach involves a semantic characterization of the VE and the user operations performed within it. Three-dimensional (3D) models used in current-generation VEs lack this information, because they are limited to geometric meshes that define the basic spatial structure and visual appearance of objects. SAVE draws on ontologies to augment the core visual information with properties about individual objects, relationships between these objects, actions that can be performed, and effects of the actions.

This ontological grounding bridges the gap between the graphical 3D models used in the VE and the instructional models used to drive learner assessment. The assessment component in SAVE analyzes semantic traces of learner actions within the VE to provide contextually relevant feedback for performance assessment.

In contrast to assessment tools that address "algorithmic" skills, which have a single or small number of acceptable responses, SAVE addresses more open-ended procedural skills that can have a range of acceptable solutions with significant variation among them.

SAVE was developed by SRI International (SRI) in the first year (Phase 1) of an anticipated two-year research and development effort within the Personal Assistant for Learning (PAL) program of the Advanced Distributed Learning Initiative (ADL).

1.2 DOCUMENT SCOPE

This document provides and a user guide (Section 2) for the Phase 1 SAVE framework, covering the perspectives of both a learner performing exercises within the system and an author developing instructional content.

Deploying SAVE in an online training domain requires the creation of domain-specific content. While SAVE provides authoring tools to support the creation of much of this content, it assumes the existence of 3D models and ontologies as a starting point. As such, this document does not cover the technical details of 3D modeling or ontology authoring.

2. USER GUIDE

This section describes how learners and content authors will use the component applications of the Phase 1 SAVE system, as shown in Figure 1. The descriptions and procedures in this user guide are based on an example exercise, Clear the Weapon, drawn from Soldier Training Publication (STP) 21-1-SMCT, Task 071-100-0032: Maintain an M16 Series Rifle/M4 Series Rifle Carbine.

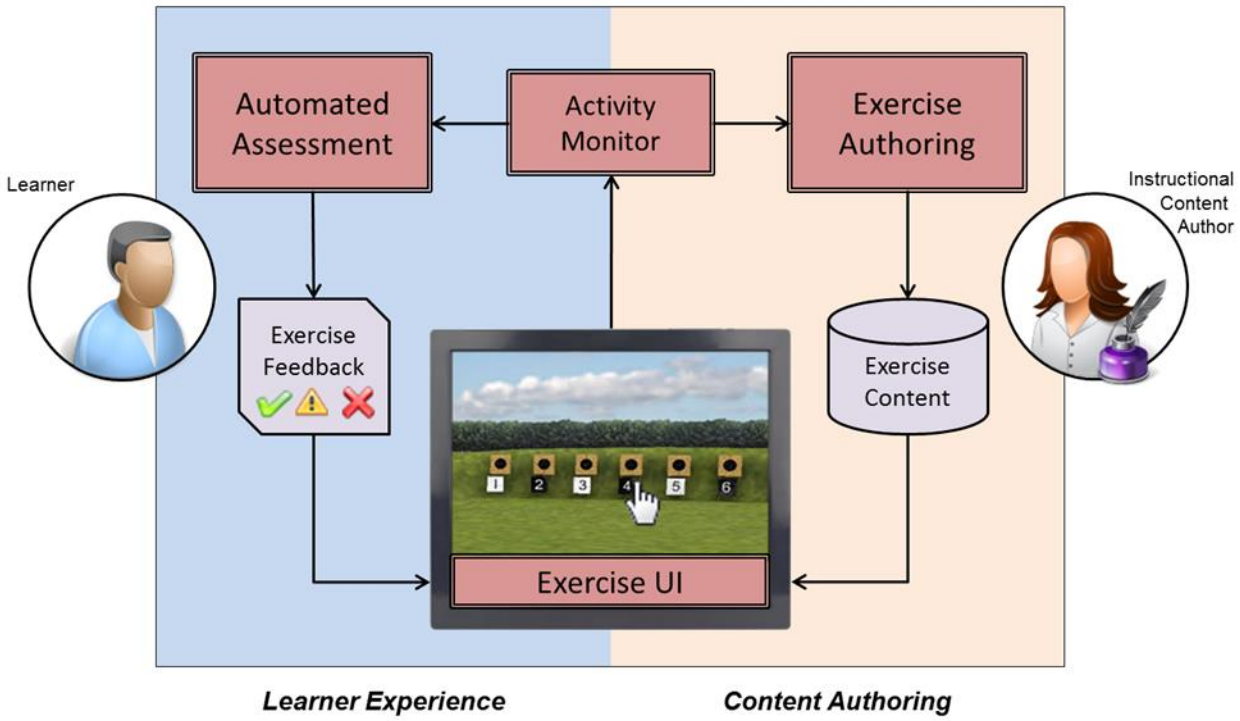


Figure 1. SAVE provides automated performance assessment and content authoring tools to support training in virtual environments.

The learner interacts with SAVE through the Exercise User Interface (EUI) to rehearse a particular skill in an interactive 3D virtual environment.

The content author creates the instructional scenario that provides skill-rehearsal opportunities to the learner. The SAVE authoring process leverages the EUI in addition to three primary content development tools:

- The Semantic 3D (S3D) annotation tool
- The Content Assembly Tool (CAT)
- The Exercise Solution Editor (ESE)

The current SAVE system is designed to support one user at a time. If multiple users access the system simultaneously, their observed activity can become intermingled during the assessment process, resulting in unexpected exercise feedback.

2.1 LEARNER PERSPECTIVE: PERFORMING AN EXERCISE

This section provides an overview of the learner interactions for performing an exercise and receiving an assessment.

Learner interactions take place within the EUI. Figure 2 shows the EUI, in learner mode, instantiated with content for the Clear the Weapon task.

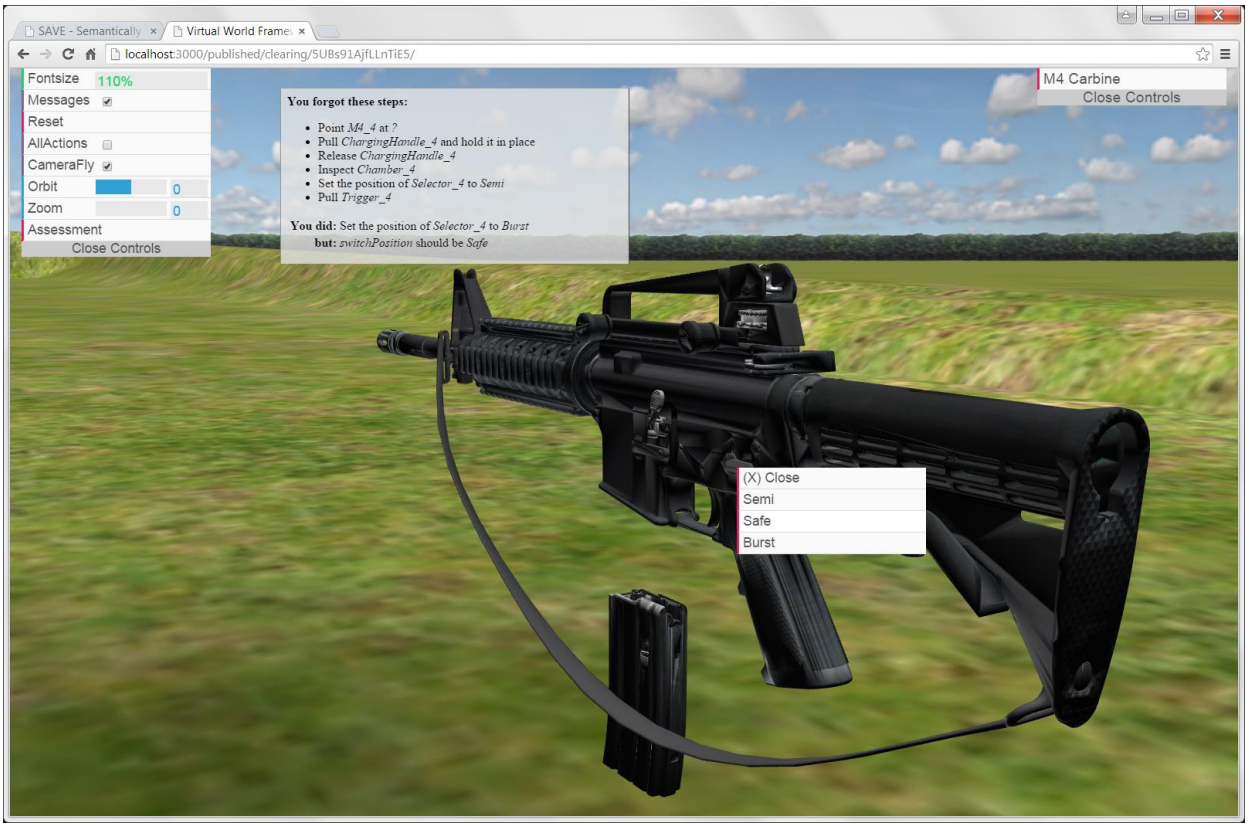


Figure 2. The EUI showing a learner assessment after completion of the Clear the Weapon exercise.

The following process describes the learner's steps for performing an exercise and receiving feedback.

1) Begin by pointing your web browser to the location of the exercise, either online or on your localhost server, if running the server on a local machine (see Section 3). Currently, these two respective URLs are as follow:

   Online: http://semantic3d.com:3000/published/clearing

   Local: http://localhost:3000/published/clearing

   The browser loads and displays the initial environment, in this case, a simple shooting range.

   NOTE: Click the Reset button once after the scene loads to refresh the assessment engine.

2) Load the M4 Carbine by selecting the M4 Carbine button in the upper-right corner.

   It can take longer than 20 seconds to load the M4 into the scene, due to the size of the high-fidelity 3D model file.

Once the loading is complete, the 3D model of the weapon is displayed front and center, and the following collection of controls is listed in the upper-left corner:

Fontsize	Changes the size of the font in the controls menu
Messages	Toggles display of the elements of the action trace as they are performed.
Reset	Reloads the screen and discards previous actions so you can start over.
AllActions	Toggles to bypass context menu filter which restricts actions are shown.
CameraFly	Toggles for freeform WASD+ mouse navigation rather than orbit/zoom.
Orbit	Provides camera control to circle around the central object, ranging from -180 to 180, to display different perspectives.
Zoom	Provides camera control to zoom close to (and away from) the central object, ranging from 0 to 90, to allow inspection of object details.
Assessment	Signals that you finished the exercise and are ready for feedback. Do not select this option until you have completed the exercise.
Close Controls	Hides the panel of user controls.

3) Initiate actions by clicking on the relevant areas of the object.

   A context-dependent menu shows the available actions for that object. Only actions that make physical sense for the object are shown, unless the "AllActions" toggle is set.

4) Select the specific action to perform.

   Steps 1a-1j for Clear the Weapon under Task 071-100-0032 of the STP, in Figure 3, document the correct process to follow for the example exercise.

   For example, step 1a says, "Point the weapon in a safe direction." Clicking on the main body of the M4 displays a context-specific menu with the option to "Position Weapon." Selecting this command rotates the weapon to point down-range. Behind the scenes, the EUI generates a message describing this action. If the Messages box was checked, the message is displayed at the top of the window. Selecting "Close" in the context menu dismisses the action without generating a message.

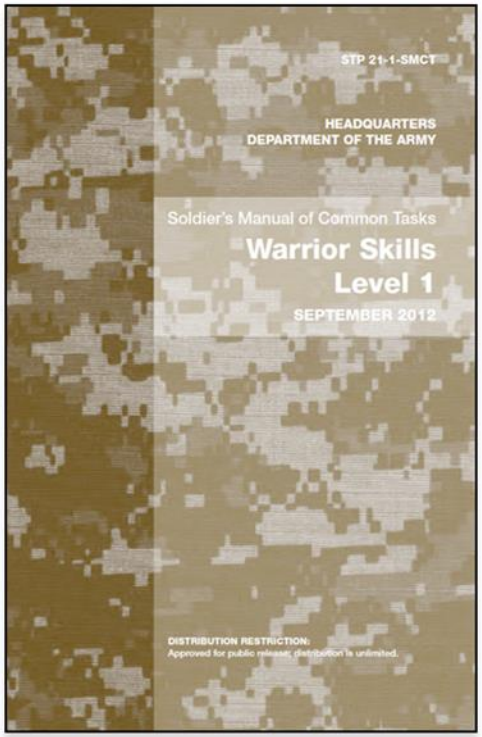
   Step 2b is to place the selector lever in *safe* mode. To obtain a better view of the selector lever, on the left-hand side of the weapon, you can change perspective by moving the orbit slider to the left. You can also look more closely at the weapon by moving the zoom slider to the right. Clicking on the selector lever presents three possible options. You select the correct response and continue through the remaining steps to complete the weapon-clearing exercise.

5) When you finish the exercise through step 2j, select the Assessment button in the upper-left of the window.

   A complete action trace describing your response is sent to the server, where it is compared to a predefined solution model. In this example, the solution model is based on Steps 1a-1j of the training document. The resulting assessment is displayed in a window in the EUI, summarizing any errors that you made.

   These can include missed steps, unnecessary steps, violations of ordering constraints, and incorrect choices (e.g., setting the mode to burst when it should have been safe).

https://armypubs.us.army.mil/doctrine/DR_pubs/dr_aa/pdf/stp21_1.pdf



Soldier Training Publication
No. 21-1-SMCT

Warrior Skills Level 1 Tasks

Subject Area 1: Shoot/Maintain, Employ, and Engage with Assigned Weapon System

071-100-0032

Maintain an M16 Series Rifle/M4 Series Rifle Carbine

Performance Steps

WARNING
Weapon must be cleared to be considered safe.

1. Clear the weapon.
   a. Point weapon in safe direction.
   b. Attempt to place the selector lever on SAFE.

Note: If weapon is not cocked, lever can't be pointed toward safe.

   c. Remove the magazine from the weapon, if present.
   d. Lock the bolt open.
      (1) Pull the charging handle rearward.
      (2) Press the bottom of the bolt catch.
      (3) Move the bolt forward until it engages the bolt catch.
      (4) Return the charging handle to the forward position.
      (5) Ensure the receiver and chamber are free of ammo.
   e. Place the selector lever on safe.
   f. Press the upper portion of the bolt catch to allow the bolt to go forward.
   g. Place the selector lever from SAFE to SEMI.
   h. Squeeze trigger.
   i. Pull the charging handle fully rearward and release it, allowing the bolt to return to the full forward position.
   j. Place the selector lever on SAFE.

STP-21-1-SMCT 11 September 2012 3-1

Task 071-100-0032 Performance Step 1a-1j: Clear the Weapon

Figure 3. STP 21-1-SMCT served as a reference for the procedural steps of the example weapon-clearing exercise.

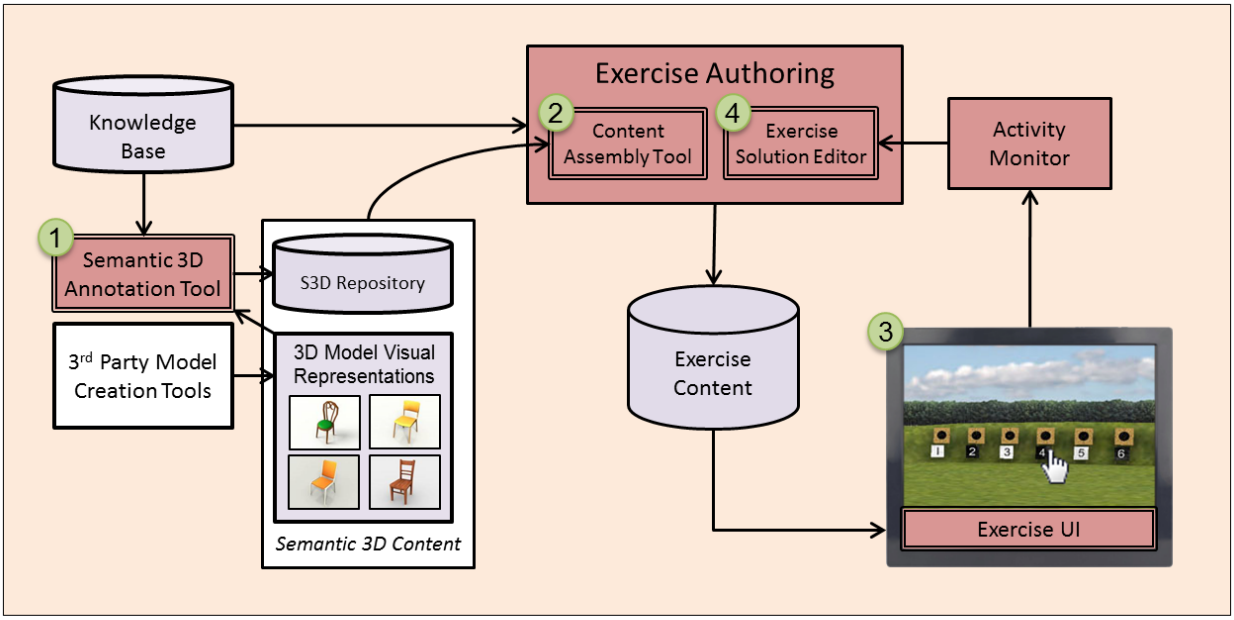
SAVE's automated assessment capability works by aligning the actions in the learner's response to a pre-defined solution model, treating deviations as errors. If the learner's response differs significantly from the solution model and the alignment is hard to determine, the assessment feedback may be unintuitive. Therefore, it is recommended that when experimenting with the system, you explore responses that have some mistakes but are reasonable attempts to complete the exercise.

2.2 AUTHOR PERSPECTIVE: CREATING CONTENT IN SAVE

Content authoring for the SAVE system covers two general areas:

- Semantic preparation of individual 3D objects that can be reused in various exercises.
- Development of a specific exercise that consists of
  - An interactive 3D scene
  - A problem definition
  - A generalized solution

The workflow (Figure 4) for the second area, exercise development, currently consists of using a series of separate tools in sequence. Future development in Phase 2 of the project will combine these tools into a more integrated authoring experience.



1. Existing 3D models are semantically annotated, i.e. ontology classes are mapped to individual nodes of the model.
2. Selected content is assembled into a 3D scene for an exercise. Additional items are identified for the tool shelf.
3. The assembled exercise is loaded into the Exercise UI and the initial solution is demonstrated in real-time.
4. The demonstrated solution is generalized within the Exercise Solution Editor by adding various annotations.

Figure 4. SAVE content authoring tools and workflow.

The following sections describe the various tools and models that support the authoring process within SAVE.

Section 2.2.1	Predefined assets that serve as the starting point for authoring
Section 2.2.2	The S3D annotation tool, which creates semantic links for 3D model objects
Section 2.2.3	The CAT for composing 3D objects into an initial scene for an exercise
Section 2.2.4	The EUI and the ESE, used in combination to author solution models for an exercise

2.2.1 Predefined Assets and Repositories

The SAVE authoring process assumes two types of pre-existing assets:

3D models:	Geometric representations of physical objects
Ontologies:	Models defining semantic properties of those objects

Existing 3D model assets are stored in the COLLADA interchange file format while ontological knowledge is represented in a framework called Flora.

For example, the Clear the Weapon exercise uses the following assets:

- Existing 3D model assets in COLLADA format:
  - M4_noHierarchy.dae
  - ShootingRange.dae
- Existing ontologies in Flora format:
  - M4.flr[1]
  - upper level ontologies: upper.flr, actions.flr, mechanics.flr

The development of these assets is the domain of other production tools and outside the scope of this document.

These two types of predefined assets are stored in individual SAVE repositories. In addition, repositories are used to store the S3D files and exercise files that are created during the authoring process. The repositories can be stored at any location definable by a URL. When the example exercise is run on the current live server, the location of each repository is as follows:

3D Models:	http://semantic3d.com:3000/SAVE/models/weapons/M4/[2]
Flora Ontologies:	http://semantic3d.com:3001/knowledge/weapons/M4/
S3D files:	http://semantic3d.com:3001/s3d/weapons/M4/
Exercise files:	http://semantic3d.com:3001/exercises/071-100-0032/step01/

[1] In Phase 1, the simple shooting range ontology was contained within the M4 ontology. This demonstrates that semantic information can be encompassed by a single Flora file, or across a number of files.

[2] This link will return an error message in your browser because the assets served on port 3000 can only be viewed as part of a composed scene in the EUI (i.e., the 3D models are intended to be accessed through programmatic means only).

2.2.2 Semantic 3D Annotation Tool

The S3D annotation tool enables a 3D content author to associate semantic classes with the corresponding nodes of a 3D model through a graphical interface (Figure 5). This association creates a link between visual representations of an object's components and background semantic knowledge that describes the object and its properties. The S3D author is assumed to be a technical artist who is familiar with 3D models and how they are put together.

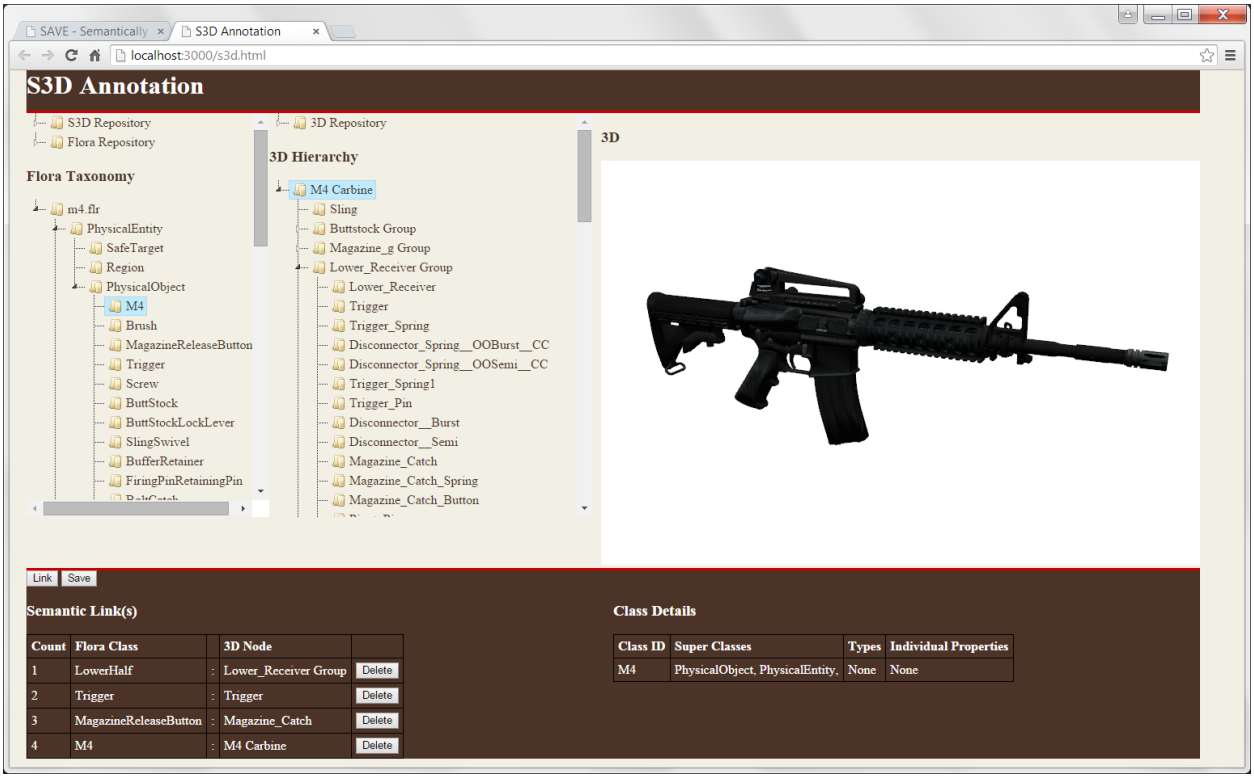


Figure 5. The browser-based UI for the S3D annotation tool.

The following instructions describe the author's process for using the S3D annotation tool to link 3D objects to ontological knowledge within the M4 domain.

1) Launch the S3D tool by pointing the browser to the location of its server, currently: http://www.semantic3d.com:3000/s3d.html

   Load the example M4 ontology file (M4.flr) from the Flora repository (see Section 2.2.1). The Flora class taxonomy is displayed in the first pane. Although many classes are shown, PhysicalObject and Region (both under PhysicalEntity), along with their numerous subclasses, are the ones of interest for annotating the 3D model.

   Load the example 3D model of the M4 Carbine (M4_noHierarchy.dae) from the 3D model repository. The 3D model is rendered on the right side and its hierarchy is displayed in the pane immediately to its left.

   Alternatively, if an S3D annotation was previously saved (e.g. M4.s3d), that may be loaded from the S3D repository menu. The Flora taxonomy and 3D hierarchy will automatically populate, and work may continue from that point.

2) To link a Flora class to a 3D node:
   a) Select one Flora class and one 3D node from the two left panes. A class can be expanded by clicking on the small arrow to its left, which exposes any subclasses. 3D nodes can be expanded in the same manner. In future versions, authors will also be able to select the node of interest in the 3D rendering window.
   b) Either click the Link button below the panes, or right-click on one of the nodes and select Link from the menu that appears. As they are made, a list of semantic links is displayed at the bottom of the window.
3) To remove linkages, either click on the Delete button or right-click on the 3D hierarchy node and selecting Unlink.
4) To examine class properties, right-click on the class of interest in the Flora taxonomy and select Info. The details will appear in the lower-right corner of the browser.
5) In the future, you will be able to select a collection of objects and create a group, which will allow you to manage subassemblies.
6) Select the Save button, above the list of links, to write the .s3d file (Section 6.6) to the repository.

2.2.3 Content Assembly Tool

The CAT allows you to build a 3D exercise scenario for an interactive training exercise. You can load previously annotated 3D models (in the S3D annotation tool) and place them in the scene or on a tool shelf for learners to access during the exercise. The CAT author is assumed to be an instructor or subject matter expert with the instructional material and objectives, who understands how to arrange a scenario to support a training exercise.

The following instructions describe how to use the CAT to compose an initial exercise scene for the M4 domain.

1) Launch the CAT by pointing the browser to the location of its server, currently:
   http://www.semantic3d.com:3000/CAT
   The CAT displays any semantically annotated 3D assets available in the repository (i.e., models with associated .s3d files)in the upper right corner.

2) Select and load the example background asset: Shooting Range.
   The shooting range environment is rendered and a menu specific to this object is displayed (Figure 6) in the upper left. Click “Shooting Range” to expand the asset options.

3) Ensure that the following checkboxes are selected:
   visible: sets the object to be visible in the exercise (default)
   auto load: sets the object to load immediately when the exercise begins

4) Select and load the example asset: M4 Carbine.
   The detailed model takes several seconds to load, and is rendered in its default location and orientation, on its side. A menu specific to the M4 Carbine is displayed (Figure 7) in the upper left. Click “M4 Carbine” to expand the asset options.

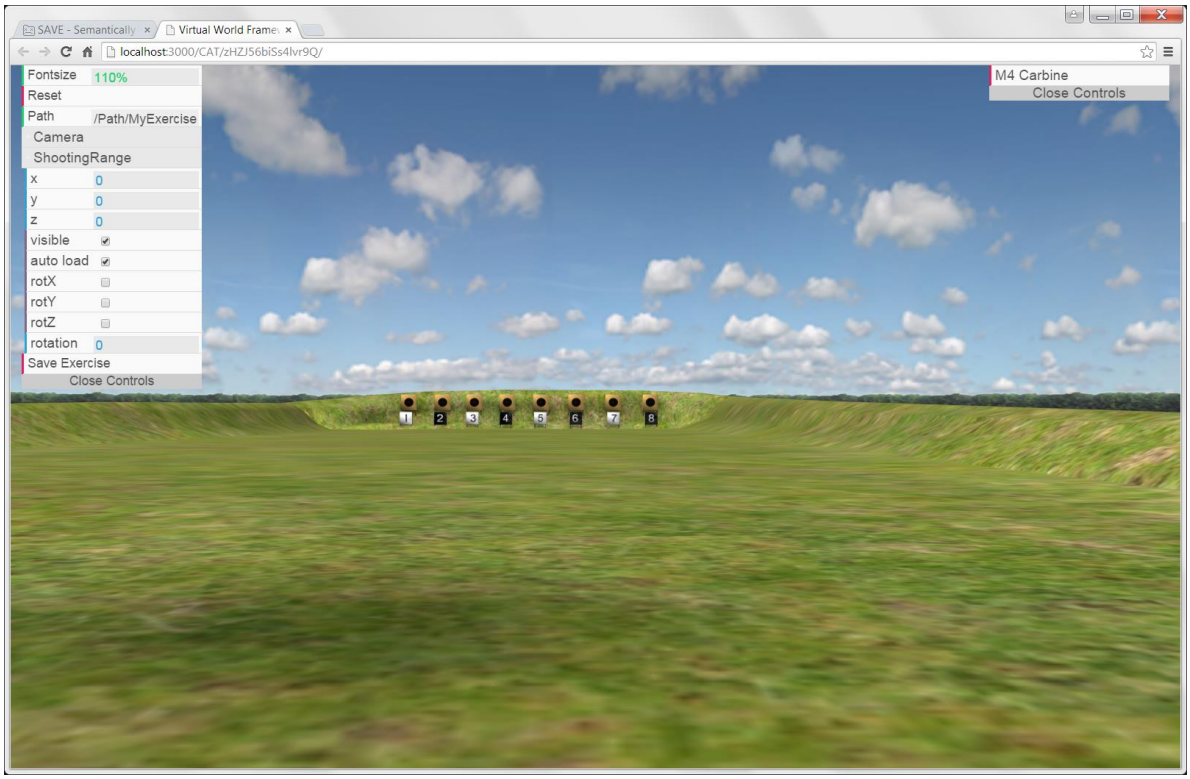


Figure 6. The browser-based user interface for the CAT with a background environment model for a new exercise scenario.

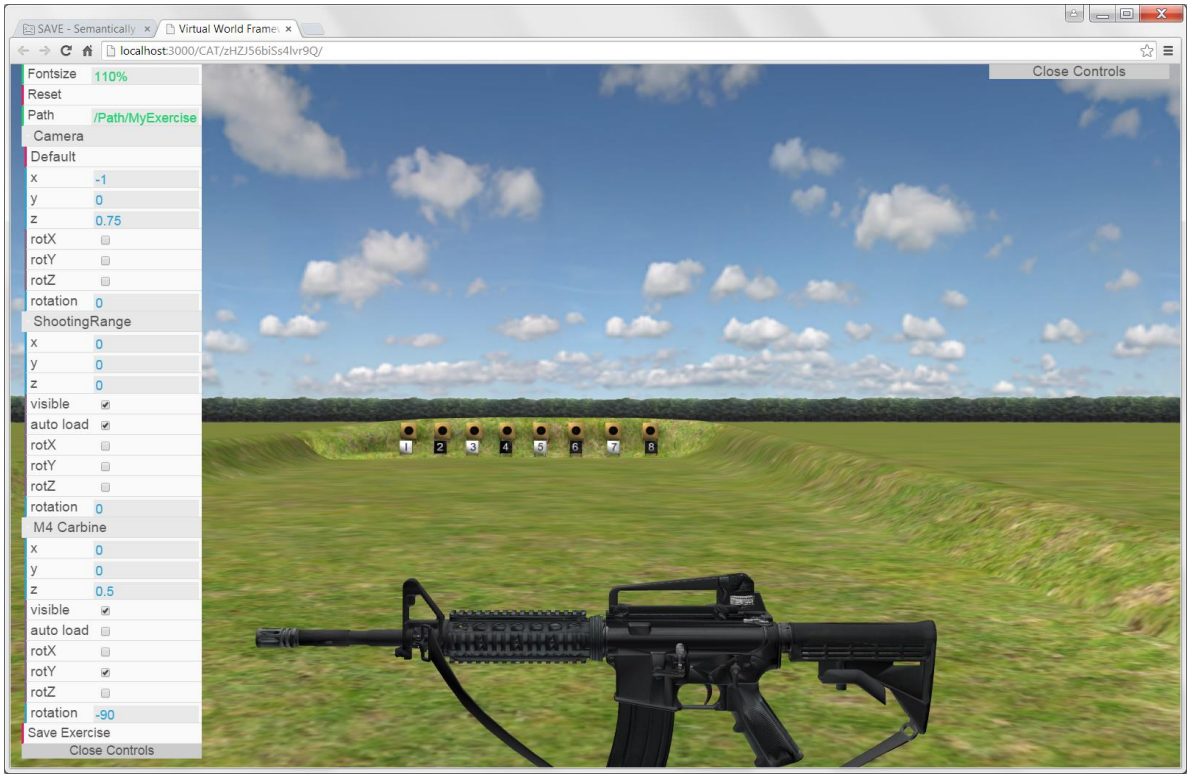


Figure 7. The CAT with a new object loaded into an exercise scenario.

5) Enter the number 0.5 in the Z field under the M4 Carbine menu.

   This moves the M4 one-half unit up, along the Z-axis (up), to position the M4 model at the desired location in the scene.

   The weapon is still lying on its side by default, but this exercise calls for the weapon to be upright and pointing 90 degrees to the left.

6) Rotate the M4 model around one axis at a time to place it in the proper orientation:
   a) Check the box next to rotZ to rotate around the Z-axis.
   b) Enter the number 90 in the rotation field and then hit enter to rotate around the Z-axis by 90 degrees. This number is a relative offset, not an absolute value.
   c) Deselect the Z-axis. Then rotate around the Y-axis, using the value of -90 (negative).

   The weapon should now be upright and pointing to the left.

7) Move the camera higher by entering the number 0.75 in the Z field under the Camera menu.

8) Enter the path and exercise name in the "Path" field. This is a path, such as /thepath/name, for example. After the path/name is entered, select Save Exercise to write the exercise to the repository. Authentication is required to write to the server, so you will need to enter a user name and password. The path appears as a button on the bottom of the left menu. The saved exercise consists of the exercise definition (Section 6.4) and VWF files (Section 6.7).

The point of view can be modified in one of two ways. The keyboard arrows (or the game-style WASD keys) allow navigation through the scene, while the mouse will change view orientation. Alternatively, the user can enter numeric values in the fields of the Camera menu in a manner similar to that used to move objects. The Default button under the Camera menu will reset the camera. The Reset button in the upper-left will reload the entire scene.

2.2.4 Exercise Solution Specification

Exercise solutions within SAVE are defined in terms of *generalized action traces*, which consist of a *sequence of steps* to be performed and *annotations* that show the allowed variability for those steps.

An instructional content author creates the exercise solution specification in a two-part process.

1) Demonstrates a solution to the exercise in the instructor mode of the EUI.
2) Uses the ESE to specify annotations that generalize from the specifics of the demonstrated solution to the full range of allowed solutions.

These two parts are described in further detail below.

The SAVE assessment framework supports solutions that can be specified in terms of generalizations of multiple demonstrations. The current UI for the system does not support tracking multiple demonstration traces; this functionality will be added to SAVE in Phase 2.

2.2.4.1 Solution Demonstration (in the EUI)

Demonstration of the solution by a content developer occurs in the EUI running in instructor mode. Author and learner mode work similarly in that both enable the user to manipulate and

perform actions on 3D objects. In instructor mode, the actions are recorded to serve as the basis for a solution; in learner mode, actions are recorded for assessment.

Instructor mode is automatic when a user accesses a fresh path saved by the CAT. The EUI will recognize that a solution has not yet been created for the exercise and will launch in instructor mode. This will be apparent by a red banner at the top, labeled "Instructor" along with the path opened. The EUI will launch in learner mode once a solution has been demonstrated and saved.

The following instructions describe how to use the EUI to create an exercise solution specification.

1) In the CAT, click the button with the link to the path link supplied in step 7 of Section 2.2.3 above. Alternatively, manually enter the URL to the path in your browser, for example: http://www.semantic3d.com:3000/thepath/name/
2) Hit the Reset button to refresh the assessment engine. Perform the exercise steps in the EUI for Clear the Weapon, in the correct order. If you make a mistake during the exercise, hit Reset again to start over. Undo/Redo capabilities are planned for implementation in Phase 2.
3) When done, select the Save Solution button to write the exercise solution to the repository.

2.2.4.2 Annotating the Demonstrated Solution (in the ESE)

The ESE enables the content author to view demonstrated action traces and add annotations to the action traces to capture allowed generalizations.

To launch the ESE:

1) Open a Command Prompt from the Windows start button via the Search programs field.
2) Navigate to the directory containing the ESE, using the command:
   ```
   cd <tasklearning root directory>
   ```
   Note that `<tasklearning root directory>` identifies the location where the tasklearning repository was checked out when setting up the ESE (Section 3.3.2.4).
3) Enter the following command to start up the ESE:
   ```
   gradlew pal-ui:ese:run
   ```
   A series of messages appears on the screen, similar to Figure 8:

   When the process completes, the ESE is displayed, although without any initial content.

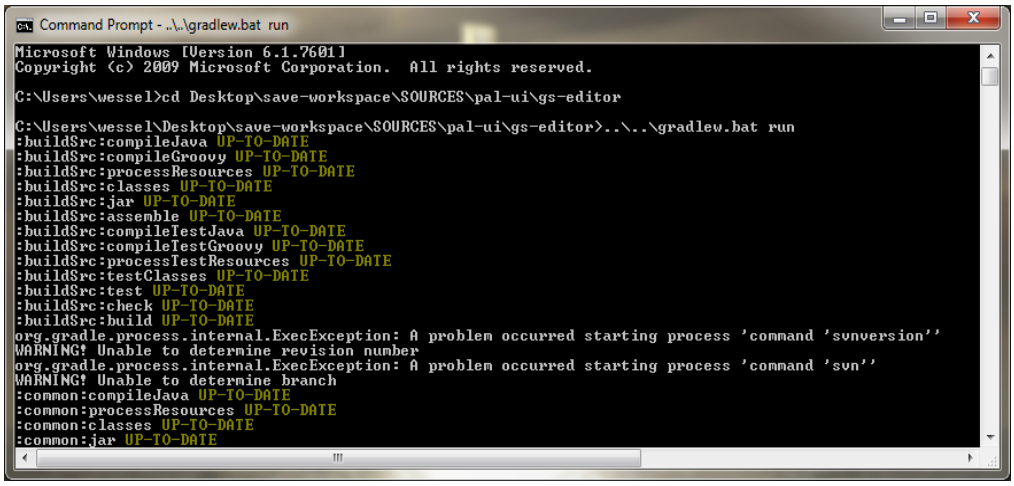


Figure 8. Windows Command Prompt showing status as the ESE is loading.

4) Select and load the Clear the Weapon solution, previously written to the SAVE repository:
   a) Use the **Open URL** button and select the appropriate URL for loading, for example: http://semantic3d.com:3001/exercises/071-100-0032/step01/m4_flora_clear_exer.xml

   The ESE displays the generalized action trace from the content developer's solution demonstration (Figure 9).

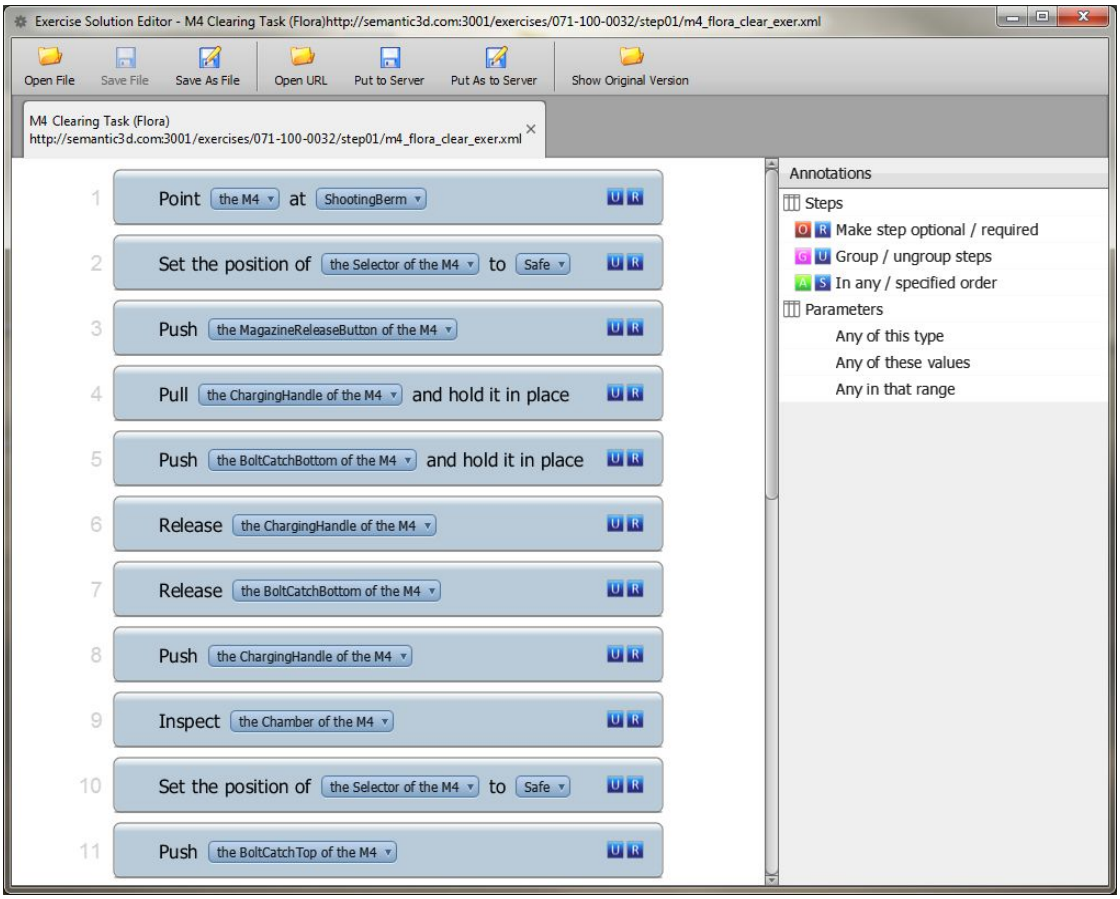


Figure 9. ESE UI showing the M4 clearing task.
The main content area shows the steps of the solution and their parameters.

5) Add annotations to the solution to indicate allowed generalities (see Section 2.3 for detailed information about using the ESE).

6) Click on the **Put As to Server** button and specify a new file name to write a revised exercise file to the repository.

2.2.5 Resulting Files in Repositories:

At the end of the SAVE authoring process, the content repository will contain the following files to represent the authored exercise:

- Existing (unmodified) 3D model assets in COLLADA format:
  - M4_noHierarchy.dae
  - ShootingRange.dae

- Existing (unmodified) ontologies in Flora format:
  - M4.flr
  - Upper level ontologies
- New semantic annotation files in .s3d format (authored via the S3D annotation tool):
  - M4.s3d
  - ShootingRange.s3d
- New Virtual World Framework (VWF) files that define the Clear the weapon exercise in the EUI (authored via the CAT tool)
- New exercise solution file (exercise.xml) (demonstrated in EUI, annotated in the ESE)

2.3 EXERCISE SOLUTION EDITOR, DETAILED USAGE INSTRUCTIONS

2.3.1 User Interface

The ESE UI has three main areas: the toolbar, the annotations panel, and the main panel (see Figure 10). The following sections provide detailed descriptions for each of these areas.

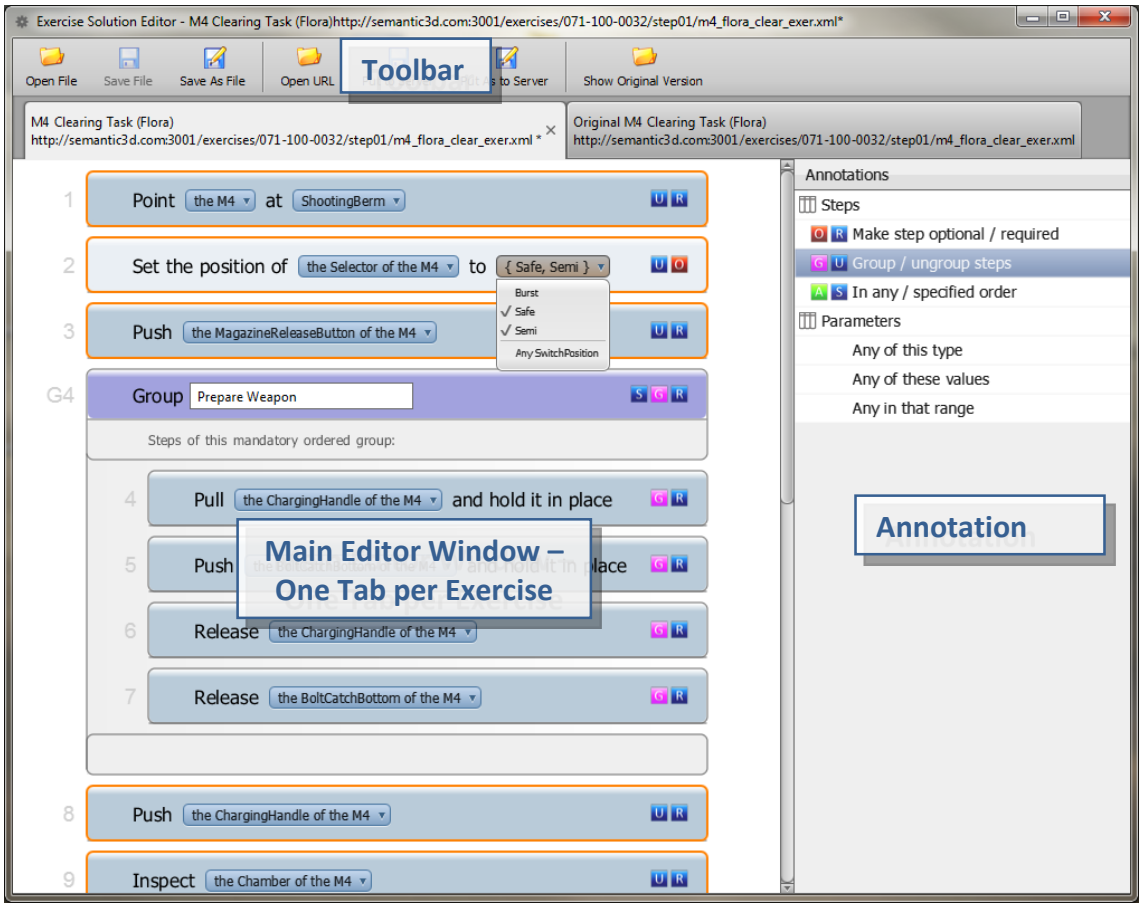


Figure 10. ESE UI.

2.3.1.1 Toolbar

The toolbar contains push-button commands grouped according to functionality.

- The Open File, Save, and Save As File commands allow you to specify input and output exercise files from the local disk. The Save As File command allows the specification of a new file name and can be used to make a copy of an exercise. *Note: the Save command overwrites the current file of the same name, without warning.*
- The Open URL, Put to Server, and Put As to Server commands allow you to input and output exercise files from an HTTP server. The current version provides a dialog box in which to enter a URL for the file to be opened or put. (A full-fledged directory browser for server content will be implemented in a future version of the ESE.) Similar to the Save As and Save commands, Put As to Server allows you to make a copy of an exercise file under a new name on the server, and Put to Server overwrites the file at the same URL.
- The Show Original Version command allows you to view the original demonstrated solution for the exercise, which can be helpful for comparison after changes have been made. The original version is loaded into a new tab with the same name as the current exercise, prefixed by Original (Figure 11). This tab is read-only to prevent accidental overwriting. Parameter values that have been changed from the original demonstration are shown in a different color to make them easy to identify (i.e., the value Burst in Figure 11).

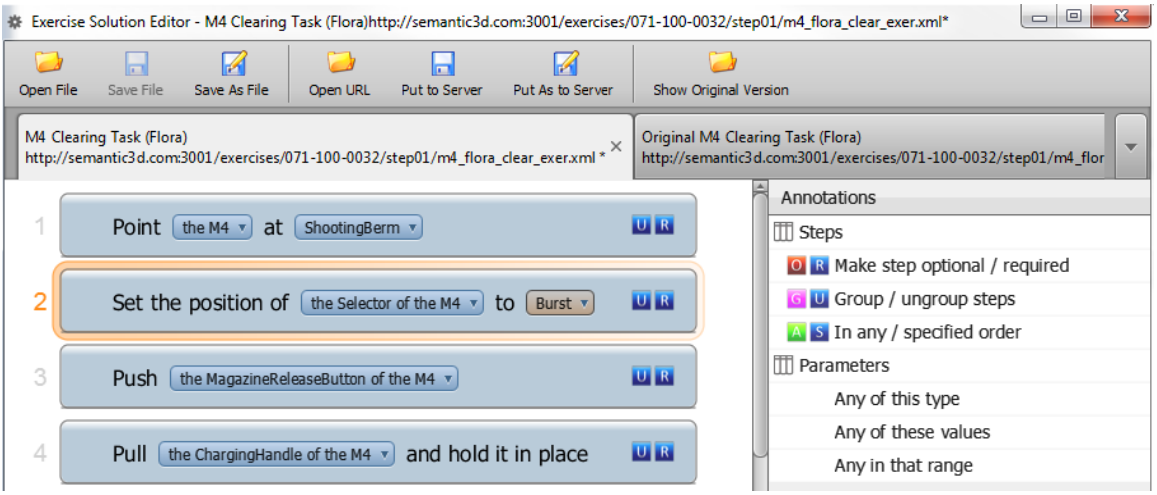


Figure 11. Display showing both the current and original solution models.

2.3.1.2 Main Panel

The main panel of ESE supports the display of multiple solution models, each within its own tab. A typical initial exercise solution trace loaded from the EUI looks similar to the example in Figure 12.

Figure 12. Original demonstrated exercise in the main panel of the ESE.

By default, the steps in an initial demonstrated solution are treated as required, ungrouped, and strictly ordered, for the purpose of assessment. Content authors use the ESE to annotate and reorder the steps, in order to define the broader set of allowed solutions for an exercise. An annotated version of this exercise might (at some point in the annotation process) look similar to the example in Figure 13.

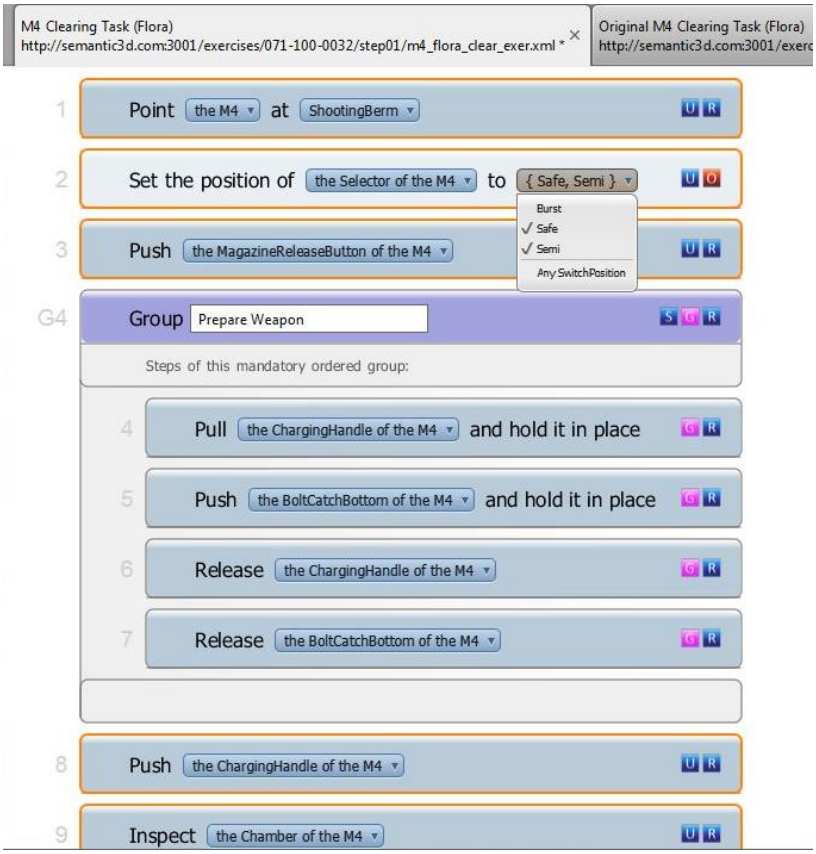


Figure 13. Annotated exercise.

2.3.1.3 Annotations Panel

The Annotations panel serves two purposes.

- It provides a legend for the annotation icons and their meanings. The same icons are used on the steps in the main window to show the annotation status.
- It provides a menu of prefix commands that set ESE into specific annotation modes. Prefix command mode requires first selecting the command and then selecting the operand(s) to which the command applies.

Annotations can be applied to steps or parameters, as described further in the sections below.

Annotating Steps

Figure 14 – Figure 16 illustrate how to annotate steps using the commands in the Annotations panel.

1) Select (for example) Make step optional / required from the Annotations panel (Figure 14). This selection allows you to toggle the required vs. optional status of steps you will select in the next step.

   ESE highlights the steps to which you can apply the selected type of annotation with an orange border, as shown in Figure 15.

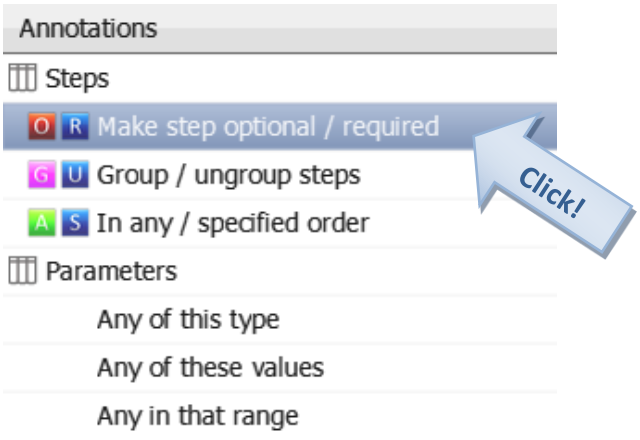


Figure 14. Annotating a step.

Figure 15. Highlighted steps that can be annotated with the selected command.

2) Click on a highlighted step in the main window (Figure 16).

   Clicking on a highlighted step toggles the annotation between required and optional. The annotation status is indicated by the color and lettering of the associated icon on that step. For required / optional, a blue R stands for required, and a red O stands for optional.

Figure 16. Selecting a step for annotation.

3) Click on the command Make step optional / required in the Annotations panel to deactivate that command.

These same basic steps apply for the remaining commands within the Annotations panel (i.e., Group / ungroup steps, In any / specified order). You can also edit a step's annotations by using the annotation icons as toggle buttons (refer to the Main Panel section).

Certain annotations are applicable to specific types of steps. For example, the In any / specified order annotation is only applicable to groups. Specifying groups and applying the In any order annotation are described in the Groups section.

Annotating Parameters

The following instructions describe how to annotate parameters using the annotation panel.

The Parameters section of the Annotations panel lists the options: Any of this type, Any of these values, and Any in that range. These commands serve as filters, highlighting the step parameters for which the annotation is possible. The following steps describe how to add the Any of these values annotations; the other annotations can be added similarly.

1) Click on Any of these values (Figure 17, right).

   Doing so highlights the parameters for which multiple values can be specified. In Figure 17 (left), this applies to the Selector switch positions of the M4 machine gun:

Figure 17. Annotating parameters.

2) Click on a highlighted parameter (Figure 18).

   Doing so displays the parameter's pull-down menu of values, from which you can make multiple selections. (Due to a technical issue with JavaFX, you have to click in the middle of the button, and sometimes, two clicks are required in order to bring up the menu.)

Figure 18. Parameter values.

3) Click on the desired value(s).

4) Deactivate the mode by clicking again on the Any of these values command in the Annotations panel.

2.3.2 Types of Steps

There are two kinds of steps in an exercise: simple steps and groups.

Simple steps can carry annotations and can be part of groups. Figure 19 provides an example of two simple steps.

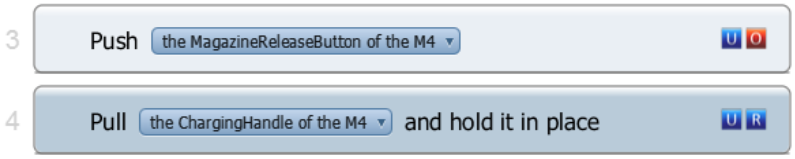


Figure 19. Annotated simple steps.

The first step, with the label 3, is an **optional** step, as indicated by the red annotation icon O. The second step, with the label 4, is a **required** step, as indicated by the blue annotation icon R. Both steps are **ungrouped**, as indicated by the annotation icon U. As described in the Annotations section, these icons act as toggle buttons. For example, clicking on the O on the step labeled 3 changes its annotation from Optional to Required. Simple steps are assumed to be ordered as presented.

The purpose of a **group** is to combine a number of simple steps that semantically belong together, i.e., a subtask such as "Prepare the Weapon." Figure 20 shows an example of a group.

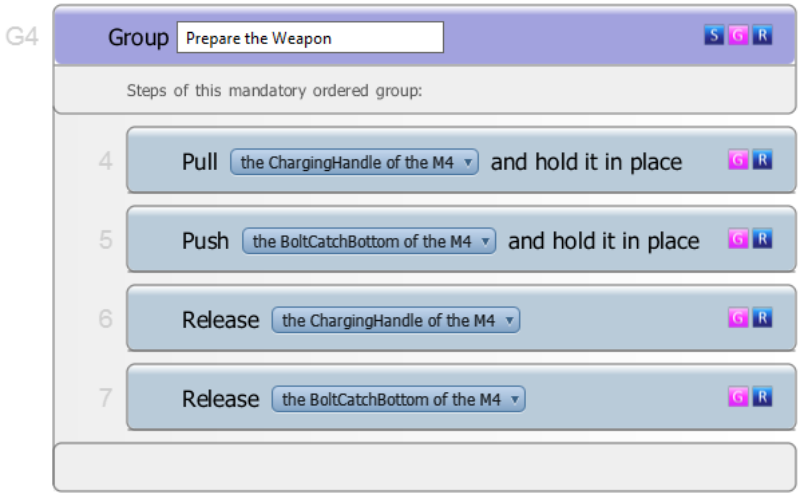


Figure 20. A group of steps for a subtask that must be performed in the specified order.

As with simple steps, steps in groups are expected to be performed in the order shown unless annotated otherwise. By using the Any Order annotation, an author can indicate that the order in which the steps of a group occur does not matter.

In Figure 21, the green icon A (in the top bar) indicates the Any Order annotation instead of the S icon (in Figure 20), which indicates that the steps must be performed in the specified order.

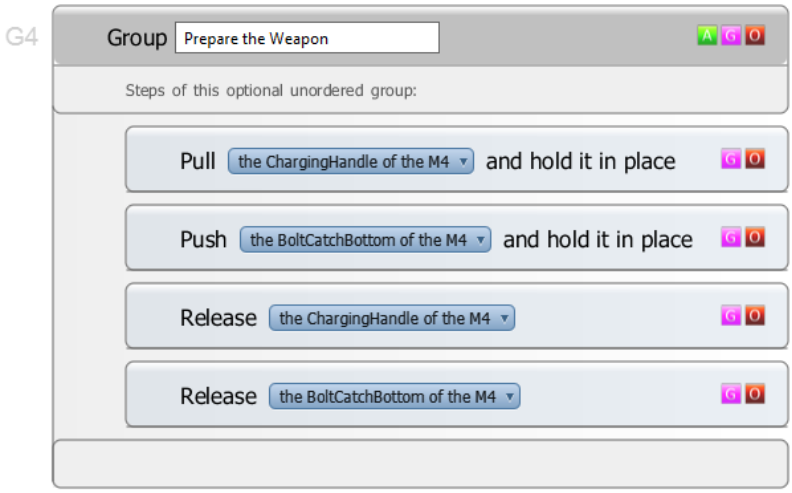


Figure 21. A group of steps that can be performed in any order.

2.3.3 Using Drag and Drop to Rearrange Steps

An author can rearrange the order of steps and groups by using a drag and drop operation. To use drag and drop (Figure 22):

1) Click on the step (or group of steps) to be moved.
2) Press and hold the mouse button.
3) Move (drag) the steps/groups to their new position and release the mouse button.

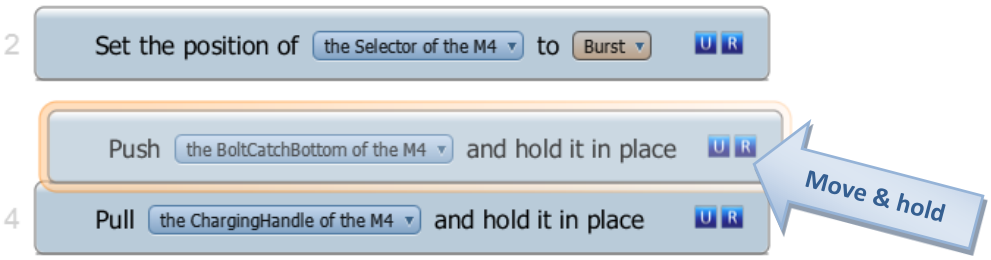


Figure 22. Using drag and drop to rearrange steps.

2.3.4 Creating and Destroying Groups

The ESE provides several methods for creating groups, as described below.

Using Group / Ungroup Parameters

If the Group / ungroup step is selected in the Annotations panel, then selecting a step that is not already part of a group creates a new group containing that step. You can enter a name for the new group in the text field on the group. As long as ESE is in this mode, you can continue to add selected (clicked-on) steps to the group. Disable the mode when the group is fully populated by unselecting the Group / ungroup steps item from the annotation panel. If you select a group while

Group / ungroup steps is selected in the Annotations panel, the group will be destroyed, i.e., its steps will become ungrouped and the group will be deleted. No steps will be deleted.

Using the U and G Icons

The U and G icons on simple steps indicate whether a step is part of a group or not (ungrouped vs. grouped). If the Group / ungroup steps annotation option is not selected, a group can also be created or destroyed by using the G and U icons on the steps and groups, respectively.

- Clicking the U icon on a step creates a new group (i.e., toggles from ungrouped to grouped). Subsequent clicks on U buttons of other steps add those to the group being created. A new (second, third, …) group can be created by enabling and disabling the Group / ungroup steps option in the annotation panel.
- Clicking the G icon on a step (part of a group) toggles the annotation to ungrouped and removes that step from the group. Removed steps are placed in front of the group. The group ceases to exist when the last step is removed from the group. Drag and Drop can be used to rearrange the position of removed steps.
- Clicking the G button on a group dissolves the group. All of its steps are removed from the group and converted back to single steps.

Using the Arguments First Method

The arguments-first method can be a natural approach to creating groups. The following instructions explain this technique.

1) Select the steps to be included in a group by using the Ctrl-Mouse click combination (press the Ctrl button and click on the required step) as shown in Figure 23.

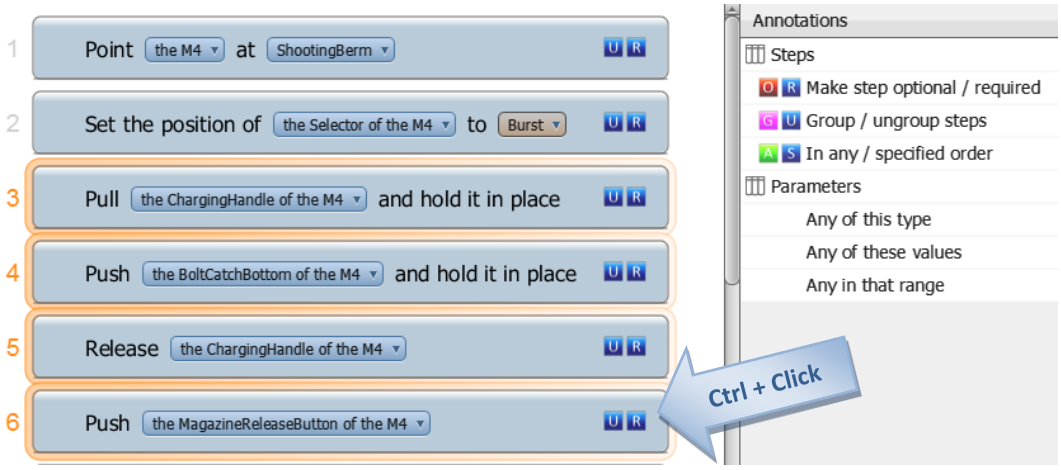


Figure 23. Selecting multiple steps.

2) Select the Group / ungroup steps command to create the group (Figure 24).

Figure 24. Creating a group from selected steps.

Adding Steps to an Existing Group

1) Make sure no menu items are selected in the annotation panel (i.e., you are not in prefix command mode).
2) Select the group you want to extend.
3) Use Ctrl-Mouse click to select additional steps to join the selected group (lines 5, 6, and 7 in Figure 25).

Figure 25. Adding steps to an existing group.

4) Select Group / ungroup steps from the annotation panel.

The steps you selected will be added to the group.

Combining Groups

The arguments-first mode can also be used to combine groups. If the selection of items contains more than one group when you select the Group / ungroup steps command, the steps from all of the groups are combined into a single group. The name of the first group in the list of selected items is used for the combined group.

You can also dissolve a group using arguments-first mode. If the current selection contains only one group (and no other groups or steps), selecting the Group / ungroup steps command dissolves this group.

3. APPENDICES

3.1 ACRONYMS

3D	Three-dimensional
AA	Automated Assessment
ADL	Advanced Distributed Learning
API	Application Programming Interface
BSD	Berkeley Software Distribution (license)
CAT	Content Assembly Tool
CDDL	Common Development and Distribution License
CLP(R)	Constraint Logic Programming (Real)
COLLADA	Collaborative Design Activity (interchange file format for 3D)
CSV	Comma-separated Values
DEFT	Drill Evaluation for Training
ESE	Exercise Solution Editor
EUI	Exercise User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JDK	Java Development Kit
JRE	Java Runtime Environment
JSON	JavaScript Object Notation
KB	Knowledge Base
KB ID	Knowledge Base Identifier
MIT	Massachusetts Institute of Technology
NURBS	Non-uniform Rational Basis Spline
OWL	Web Ontology Language
RAVE	Rule Authoring and Validation Environment
REST	Representational State Transfer
S3D	Semantic 3D
SAVE	Semantically Enabled Automated Assessment in Virtual Environments
SDK	Software Development Kit

SQL	Structured Query Language
SSG	Semantic Scene Graph
SVN	Subversion (version control system)
SWRL	Semantic Web Rule Language
UI	User Interface
URL	Uniform Resource Locator
VE	Virtual Environment
VWF	Virtual World Framework
XML	Extensible Markup Language

3.2 SOFTWARE COMPONENT LICENSES

SAVE software components and other SRI software are not currently licensed for distribution, unless otherwise stated. Third-party software is governed by the license under which it is distributed.

The United States Government retains unlimited rights (See DFARS 252.227-7013 and 7014) to the software and technical data developed under the contract that are not otherwise designated as "special works" by the Contracting Officer, with "special works" as defined by DFARS 252.227-7020.

3.2.1 SAVE Software Components

Component:	**SAVE Backend (SAVE)**
Copyright:	SRI International
License:	TBD
Used by:	n/a
Comment:	

Component:	**SAVE Exercise UI (EUI)**
Copyright:	SRI International
License:	TBD
Used by:	n/a
Comment:	

Component:	**SAVE S3D Annotation Tool (S3D)**
Copyright:	SRI International
License:	TBD
Used by:	n/a
Comment:	

Component:	**SAVE Exercise Solution Editor (ESE)**
Copyright:	SRI International
License:	TBD
Used by:	n/a
Comment:	

Component:	**SAVE Content**
Copyright:	SRI International
License:	TBD
Used by:	EUI, S3D, ESE
Comment:	Does not pertain to any public domain asset(s) used.

3.2.2 Other SRI Software Used by SAVE

Component:	**Drill Evaluation for Training (DEFT)**
Copyright:	SRI International

License: TBD
Used by: SAVE
Comment: Approved by U.S. Government for public release.

Component: Adept
Copyright: SRI International
License: TBD
Used by: DEFT
Comment: Approved by U.S. Government for public release.

Component: Sunflower Foundation (formerly FloraLib)
Copyright: SRI International
License: Sunflower Foundation End User License Agreement
Used by: SAVE, S3D
Comment: Distribution A; binary only.

Component: floralib-ext
Copyright: SRI International
License: LGPL
Used by: Sunflower Foundation
Comment:

3.2.3 Third-party Software Used by SAVE

All libraries listed here are unmodified, unless otherwise stated.

Component: Virtual World Framework
Copyright: U.S. Government
License: Apache 2.0
Used by: SAVE, EUI, S3D
Comment:

Component: Gson
Copyright: Google Inc.
License: Apache 2.0
Used by: SAVE
Comment:

Component: dat-gui
Copyright: Data Arts Team, Google Creative Lab
License: Apache 2.0
Used by: EUI
Comment:

Component: sax-js
Copyright: Isaac Z. Schlueter
License: BSD
Used by: EUI (nodejs)
Comment: Contains code with MIT license

Component: JSTree

Copyright: Ivan Bozhanov
License: MIT
Used by: S3D
Comment:

Component: Jquery
Copyright: jQuery Foundation
License: Jquery
Used by: S3D
Comment:

Component: jaxb-impl
Copyright: Oracle Corporation
License: CDDL 1.1 or GPL2
Used by: DEFT
Comment: Dual-license, SRI uses under CDDL

Component: Scala
Copyright: EPFL and Typesafe, Inc.
License: Scala
Used by: DEFT
Comment:

Component: ActiveMQ
Copyright: Apache Software Foundation
License: Apache 2.0
Used by: DEFT

Component: Velocity
Copyright: Apache Software Foundation
License: Apache 2.0
Used by: DEFT
Comment:

Component: commons-collections
Copyright: Apache Software Foundation
License: Apache 2.0
Used by: Velocity, Sunflower Foundation
Comment:

Component: commons-lang
Copyright: Apache Software Foundation
License: Apache 2.0
Used by: Velocity, Sunflower Foundation
Comment:

Component: commons-codec
Copyright: Apache Software Foundation
License: Apache 2.0
Used by: Sunflower Foundation

Comment:

Component: **commons-configuration**
Copyright: Apache Software Foundation
License: Apache 2.0
Used by: Sunflower Foundation
Comment:

Component: **commons-csv**
Copyright: Apache Software Foundation
License: Apache 2.0
Used by: Sunflower Foundation
Comment:

Component: **commons-io**
Copyright: Apache Software Foundation
License: Apache 2.0
Used by: Sunflower Foundation
Comment:

Component: **commons-logging**
Copyright: Apache Software Foundation
License: Apache 2.0
Used by: Sunflower Foundation
Comment:

Component: **httpcomponents-client**
Copyright: Apache Software Foundation
License: Apache 2.0
Used by: Sunflower Foundation
Comment:

Component: **Apache Xerces 2.10.0**
Copyright: Apache Software Foundation
License: Apache 2.0
Used by: Sunflower Foundation
Comment:

Component: **h2**
Copyright: H2 Group
License: Mozilla 2.0 or Eclipse 1.0
Used by: Sunflower Foundation
Comment: Dual-license, SRI uses under Mozilla.

Component: **junit**
Copyright: Junit
License: Eclipse 1.0
Used by: Sunflower Foundation
Comment:

Component: **jackson**

Copyright:	FasterXML, LLC
License:	Apache 2.0 or LGPL 2.1
Used by:	Sunflower Foundation
Comment:	Dual-license, SRI uses under Apache

Component:	**Jena**
Copyright:	Apache Software Foundation
License:	Apache 2.0
Used by:	Sunflower Foundation
Comment:	

Component:	**Jena IRI**
Copyright:	Apache Software Foundation
License:	Apache 2.0
Used by:	Sunflower Foundation
Comment:	

Component:	**XSB**
Copyright:	The Research Foundation of the State University of New York; ECRC, Germany
License:	LGPLv2
Used by:	Sunflower Foundation
Comment:	

Component:	**Flora**
Copyright:	The Research Foundation of the State University of New York; Vulcan, Inc.
License:	Apache 2.0
Used by:	Sunflower Foundation
Comment:	

Component:	**Interprolog**
Copyright:	Declarativa/XSB Inc.
License:	LGPLv2
Used by:	Sunflower Foundation
Comment:	

Component:	**joda-time**
Copyright:	Joda.org
License:	Apache 2.0
Used by:	Sunflower Foundation
Comment:	

Component:	**slf4j**
Copyright:	QOS.ch
License:	MIT
Used by:	Multiple components
Comment:	

Component:	**log4j**
Copyright:	Apache Software Foundation

License:	Apache 2.0
Used by:	slf4j, Sunflower Foundation
Comment:	

Component:	**Jetty**
Copyright:	The Eclipse Foundation
License:	Apache 2.0 or Eclipse 1.0
Used by:	SAVE
Comment:	Dual-license, SRI uses under Apache.

Component:	**javax servlet API**
Copyright:	Oracle Corporation
License:	CDDL 1.1 or GPL2
Used by:	Jetty
Comment:	Dual-license, SRI uses under CDDL.