TaskGuide
Rapid Development of Adaptive Training and Performance Support Systems
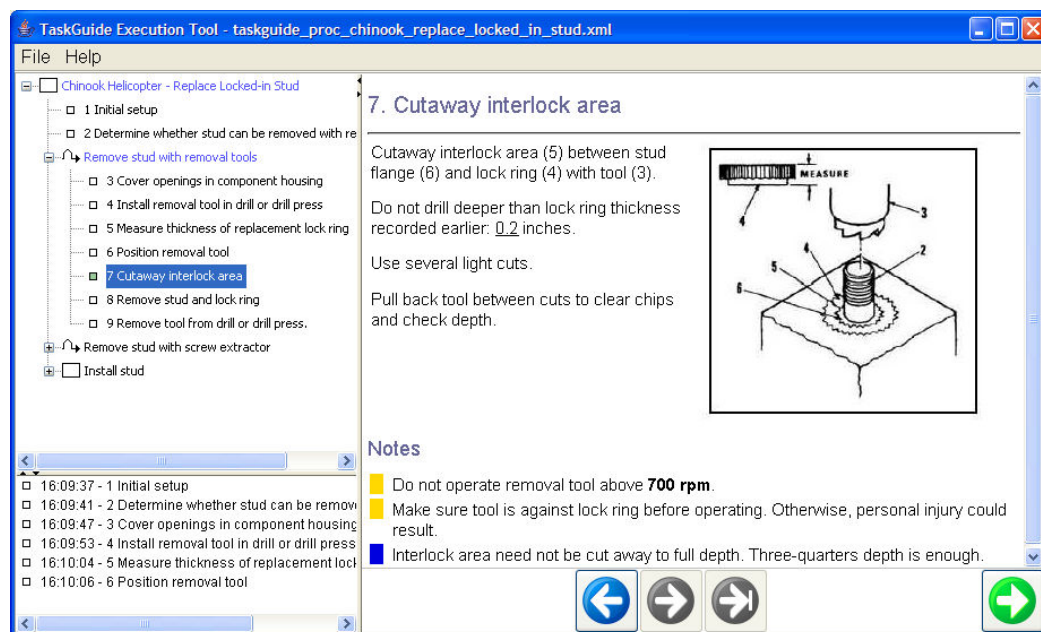
Overview

The TaskGuide™ software system enables organizations to develop sophisticated applications called procedures that retrieve, compute, update, display, and accept information using a step-by-step wizard-like user interface. TaskGuide provides a graphical authoring tool that enables authors to create these procedures quickly and easily, without programming. Powerful authoring capabilities enable you to create *adaptive* procedures that determine which information to present and which questions to ask, depending upon the situation. TaskGuide is modular and extensible: it can be embedded within a larger Java applications or applet, and it can embed Java user interface components and libraries.

TaskGuide procedures can be used to implement diverse types of applications such as:

- **Training tutorials** that present information to students, pose questions or problems, and provide tailored hints and feedback. Student performance can be tracked in files or by a SCORM learning management system.

- **Performance support systems** such as task aids and decision support systems that guide users through analysis, decision-making, and execution tasks, step-by-step.

- **Training simulations** such as branching scenarios that present the current situation in each screen, prompt the student for decision or actions, and branch to the appropriate next situation.

- **Adaptive questionnaires** that determine which questions to ask based on previous answers and other available data.

- **Debriefing systems** that analyze and discuss the student's perceptions, decisions, and actions during a simulation-based training or assessment activity. This discussion format enables the software to refine its assessment of the student's proficiencies, provide instructional feedback, and guide reflection.

TaskGuide Procedure Execution Tool

Users and students use the TaskGuide Procedure Execution Tool to run procedures. In the figure below, the procedure summary pane (upper left) shows a graphical summary of the steps and groups of steps in the procedure. The details pane (right) shows instructions and input controls for the current step. The log pane (lower left) lists each step that has been executed. The user presses the green arrow button (lower right corner) to advance to the next step. Other buttons enable the user to review previously-executed steps.

The procedure summary pane displays an icon and label for each step and group of steps. Different icons represent different types of steps and groups as shown in the table below.

	Interactive	Invisible
Simple Step	□	■
Exit Step	↵	↵
Conditional Branches Step	◇	◆

Simple Group	□
Branching Group	⌒↳
Looping Group	↻

Each simple step presents information to the user and can prompt the user for input. An exit step specifies an exit condition that, if true, instructs TaskGuide to stop executing the current group. The conditional branches step specifies many possible steps to execute next, depending upon the values of certain variables. A simple group organizes related steps and subgroups. All of the steps and sub-groups contained within a simple group are executed in sequence. A branching group contains steps that are executed only if a test condition is true. A looping group executes its steps repeatedly while a test condition is true.

A step's instructions can contain **input controls** such as text fields, check boxes, radio buttons, and selection lists that prompt the user to enter data and decisions. TaskGuide stores user input values in variables, so they can be used within the calculations and test conditions of downstream steps. TaskGuide provides special support for specifying multiple choice questions with response-specific hints and feedback. TaskGuide steps can present *verifications* that tell the user how to confirm that the step was completed successfully. Optional *notes* communicate *information, cautions,* and *warnings.* Step instructions can also embed arbitrary Java graphical user interface objects that provide specialized information displays or user-system interactions.
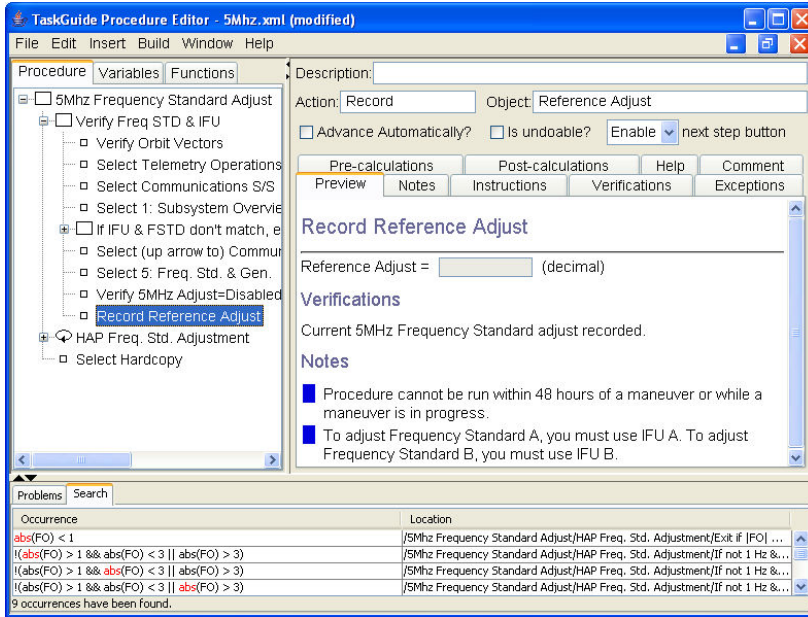
Steps can contain **calculations** that evaluate Javascript expressions and save these values in variables. These expressions can contain constant values, variables, operators, and calls to Javascript functions and Java methods. The saved variable values can be used by branching logic to control which downstream steps are executed and which are skipped. They can also be used within calculations in downstream steps to send/receive data to/from other systems and databases, analyze and interpret this data, recommend actions to be taken by the user, or select and execute actions automatically. *Pre-calculations* execute at the beginning of each step before the step's instructions are displayed to the user. They are useful for retrieving and computing data or text so they can be embedded within dynamic instructions. *Post-calculations* execute at the end of the step, after the user has followed the step's instructions, entered data into input fields, and pressed the next step button to indicate completion. They are useful for interpreting, processing, saving, or acting upon the user's inputs.

TaskGuide Procedure Editor

Authors use the TaskGuide Procedure Execution Tool to create procedures. A TaskGuide procedure encodes step-by-step instructions and execution logic as a list of steps, organized within groups and subgroups in a hierarchy. Each step presents instructions to the user using formatted text and graphics specified using the HyperText Markup Language (HTML). Instructions can contain **hyperlinks** to web pages that present additional information on demand in a web browser using text, graphics, and other media.

The TaskGuide Procedure Editor is shown below. The overview pane (left) contains tabbed windows that display the procedure's steps and groups, along with the variables and Java library functions that can be used in procedure steps. The details pane (right) lets authors edit the step or group that has been selected in the left pane. The bottom pane lists string search results and problems detected automatically by the Editor.

Each step can contain static or dynamic instructions and verifications. **Static instructions** present the same information each time the procedure specification is executed. Authors use the built-in graphical HTML editor to specify the content and format of static instructions as formatted text, images, and hyperlinks.

You can also specify **dynamic instructions** by **embedding expressions** within the instruction's HTML text. During execution, TaskGuide evaluates each embedded expression and replacing the expression with its value. Expressions often contain references to variables whose values can be entered by the user, received from external systems and databases during procedure execution, or computed from mathematical, boolean, or string expressions that refer to other variables. You can also use **conditional inclusion** to include some HTML tags and formatted text and exclude other tags and text when displaying a step, based on the values of certain procedure variables. Compared to static instructions, dynamically-generated instructions can filter information to present instructions that are more succinct and targeted to the user or situation. They can also select or compute hints, feedback, default values, or recommendations. You can also **embed Java user interface objects** to provide specialized interactive displays.

TaskGuide supports **gradual automation**, so steps in a manual procedure can be replaced, one at a time, with steps that use calculations to retrieve data, compute values, and carry out actions automatically. For example, a simple interactive step could simply provide instructions that tell the user how to carry out a particular task. The step could be enhanced with calculations that automate the retrieval of relevant data. Further enhancement might use calculations to compute default parameter values or decisions and prompt the user to confirm or override them. As confidence increases in the reliability and robustness of the automated recommendations, the organization could replace the interactive step with a fully automated step that analyzes data, decides and acts without user intervention. In this manner, a manual procedure can evolve over time into a more automated one.

TaskGuide's **extensible architecture** enables integration with general purpose and application-specific Java software libraries that provide functions that are invoked by calculations. This architecture enables TaskGuide procedures to incorporate sophisticated automated data retrieval, interpretation, automated reasoning and decision-making algorithms. The TaskGuide application programming interface (API) enables you to embed TaskGuide within a larger Java application or applet.

About Stottler Henke

Stottler Henke Associated, Inc. is an artificial intelligence software research and development company headquartered in San Mateo, CA, with offices in Seattle, WA, and Cambridge, MA. The company develops intelligent software systems for simulation and training, planning and scheduling, decision support, knowledge management, information access, and data mining. For additional information, call us at (650) 931-2700, or send email to info@stottlerhenke.com.