

STIL-Pro™

STIL-Pro - general description

STIL-Pro is a modular building block designed to be the foundation of STIL-based applications and implements all of the language capabilities of STIL and STIL extensions in a fully optimized manner. It is packaged as a shared library module with API which can easily be integrated into any application.

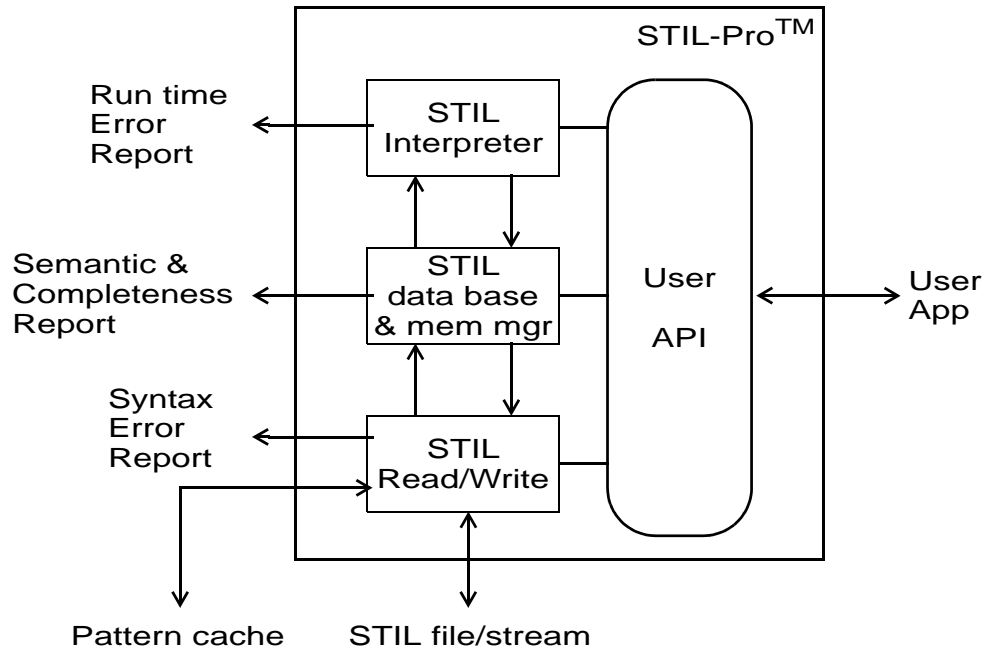
STIL-Pro - features

Basic STIL for patterns	all syntax of IEEE Std. 1450-1999
Design extensions	all syntax of P1450.1-D22
DC levels	all syntax of IEEE Std. 1450.2-2002
Core Test Language	all syntax of P1450.6-D1.5
Tester rules check *	all syntax of P1450.3-D09
Data compression *	in memory and file compression for efficiency and capacity
Disk file caching *	large volume items are cached to disk - patterns, scan structures, environment
C and C++ * API	C and C++ libraries available
Multiple platforms	available for Solaris, Linux, and HPUX
Error reporting	full checking for – syntax errors, reference errors, semantic errors
STIL interpreter	API functions return data as resolved, interpreted vector and timing data under several processing options
Reader / writer	API functions for accessing parsed STIL data with view, delete and insert capabilities
Multiple STIL context	API can manage multiple STIL file/streams
STIL domains	domain specification supports STIL data re-use (e.g. pattern mapping)
Documentation	user manual, API documentation (html), object data model (uml), example applications

* feature under development

STIL-Pro - architecture

The following diagram illustrates the main components of the software and the interfaces between the user software and STIL-Pro.



User API - is the mechanism whereby the user application software controls all activities of STIL-Pro. By invoking a stil-read operation, STIL-Pro reads, parses, and loads the STIL database. In the process of reading the STIL file, any syntax errors are reported. Once a STIL file has been loaded, the user-application can direct STIL-Pro as to how it wants the data presented, and then issues a series of access calls to the user-api to read or modify the data. An API Validate function provides a comprehensive semantic and consistency check of the parsed STIL database.

STIL Read/Write - contains functions that manage the STIL file/stream. It is responsible for reading, parsing, syntax error reporting, and populating the STIL database. Optionally, handling of large blocks of STIL data (pattern blocks and scan-structure blocks) is done by use of separate STIL include files that are parsed when needed. API functions are available to provide STIL-ASCII data and line numbers to the user application.

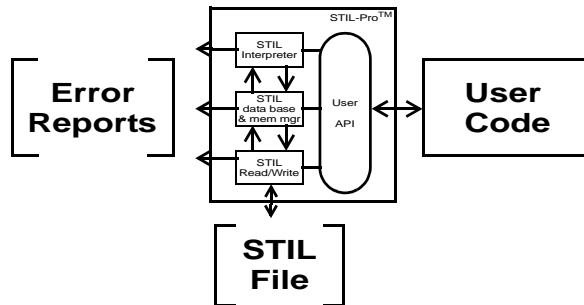
STIL data base and memory manager - is a set of parsed C Structures that are built from STIL source files. The user-api has functions to read, delete and insert at the statement and block level. Random, as well as sequential, access to STIL file data is provided. Management functions are provided to handle large files and multiple contexts.

STIL Interpreter - is a set of functions that resolve and interpret the STIL database information in accordance with the requirements of the user application. Functions provide access to resolved data and vector sequences. User options on vector processing provide control over scan data flattening, loop expansion, cyclized vs. event data streams, waveform characters vs. classical event timing formats.

STIL-Pro - applications

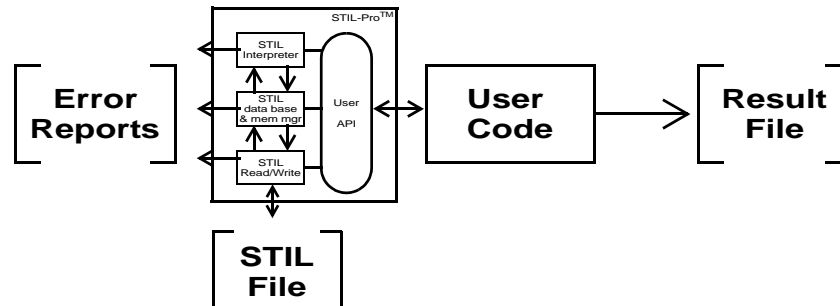
The following are a series of typical applications that use STIL-Pro to perform the tasks of reading, managing, and interpreting STIL data.

STIL-Pro used for STIL validation



The user-code, in this case, is responsible for selecting the STIL file to be validated, selecting the level of error checking desired, and invoking the user-api function to perform the checking and produce the validation report.

STIL-Pro used in a translation tool running in batch mode

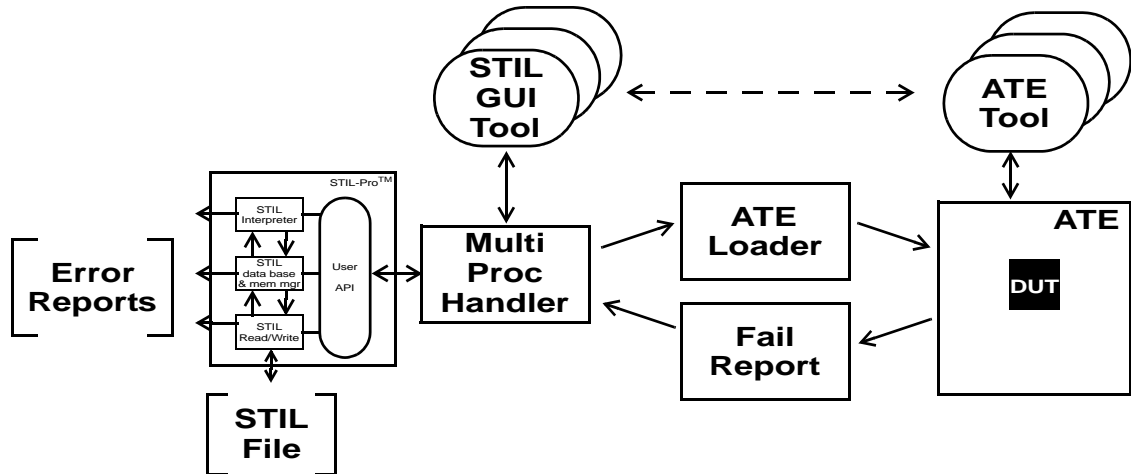


In this batch application, the user-code first invokes a call to the user-api which causes the STIL file to be read into the STIL database. In so doing, the data is checked for syntax errors, semantic errors and missing references. Any errors detected are reported and the user-code is presented an error condition.

Once the STIL database has been loaded, the user-code issues instructions through the user-api indicating the level of interpretation of the data that is desired. One option allows the user-code to traverse the STIL data by presenting the keywords and attribute data for each statement. Another option presents the data such that each statement is resolved - vectors are flattened, waveform characters are resolved to the waveform table that defines them, spec variables are resolved to the category selection, etc. Further options allow for the interpretation of shift, loop,

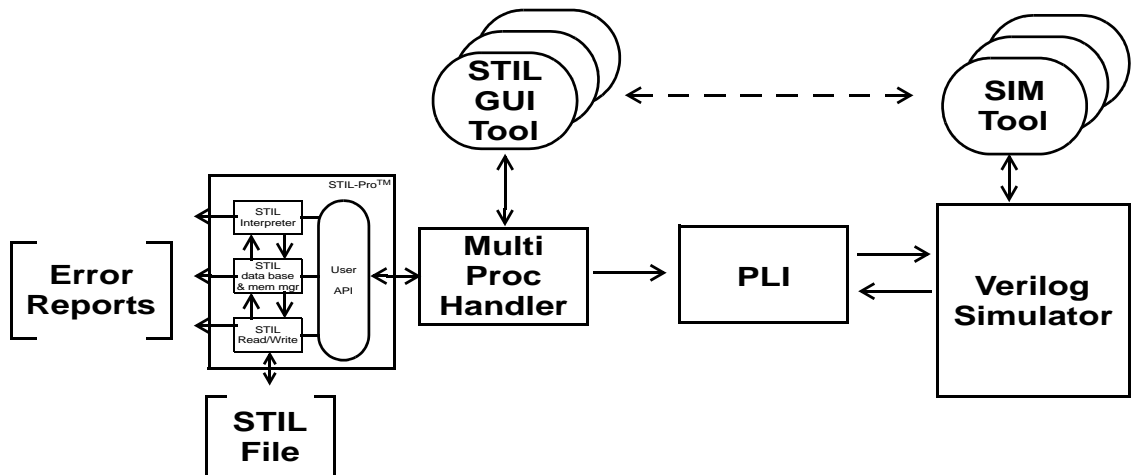
macros, procedures, etc. as required, to support the capability of the target tester application.

STIL-Pro used to run native STIL on a tester



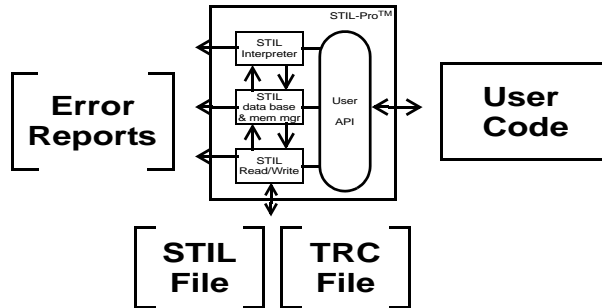
The above is a simplistic view of how STIL-Pro can function as a component of a native-STIL operating system for a tester. The multi-proc-handler block is to allow multiple processes to access the stil-data - the STIL GUI tools as well as the ATE loader and fail report.

STIL-Pro used to drive Verilog simulation



This application looks quite similar to the previous ATE application. The obvious difference is that the STIL data provides the stimulus and response to Verilog through the PLI. Another, not so obvious difference is that, whereas the tester loader would load blocks of vectors to the ATE memory, the PLI would request raw event data from STIL-Pro to schedule the drive/compare events as needed for simulation.

STIL-Pro used for tester rules checking or tester targeting



In this example, two STIL files are loaded in the STIL database - one for the test program, and one for the tester rules file. The user-code accesses the rules, then requests statistical information about the stil test program to check against the rules.

This same model also applies to a tester targeting application. In this case, the user-code block is responsible for adjusting or annotating the STIL database as needed to support a given ATE configuration. Then the write function of STIL-Pro is used to update the STIL file.

For more information contact:

Source III, Inc.

corp@sourceiii.com

916-941-9403

Stil-Pro(TM) is a trademark of Source III, Inc.

