Soar 9.3.2

Mazin Assanie, John Laird, Joseph Xu

Key Changes

- Uniform compilation procedure for all platforms compiles Soar as a single library
- New forgetting mechanism + misc. kernel changes
- Revamped and cleaned up Soar home page and Wiki
- New project organization in Wiki and SVN repository

Simplified Soar Shared Library

- ClientSML, ConnectionSML, ElementXML, KernelSML, SoarKernel combined into single Soar library

```
> g++ env.cpp -lElementXML -lClientSML -lConnectionSML -lSoarKernel
> g++ env.cpp -lSoar
```

- Static build is actually static
- No more LibraryLocation or LibraryName arguments in CreateKernelIn*Thread
- The way SpawnDebugger searches for SoarJavaDebugger.jar changed

Unified Build System

- Soar Suite now compiled via a single cross-platform SCons script
  - SCons is included in SoarSuite, you only need python
- No longer requires Ant for Java components
- No longer maintaining Visual Studio project files
  - SCons can generate one for debugging, etc
  - Still uses MSVC++ compiler
  - build.bat automatically sets all necessary environment variables
- Defaults to clang compiler on Mac, g++ 4.2 outdated
- Can compile Linux Standard Base compatible libraries

How To Compile

> svn co https://soar.googlecode.com/svn/trunk/SoarSuite
> cd SoarSuite
> python scons/scons.py

-- or if you have scons installed --

> scons

-- or for Windows --

> build.bat

Separate Compilation Targets

```
> scons list
Building intermediates to build
Installing targets to /home/jzxu/SoarSuite/out
C# compiler not found, not building C# SML wrappers
all
cli
debugger
debugger_api
headers
java_sml_misc
kernel
sml_java
sml_python
tests
> scons -c all
> scons kernel debugger headers
```

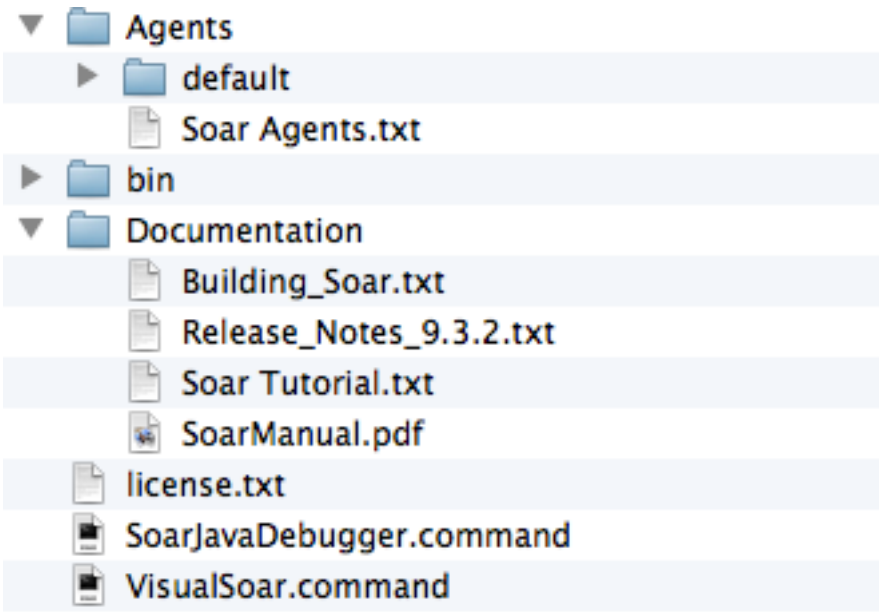
Kernel changes

- New forgetting mechanism to automatically excise chunks via a base-level decay model (look at rl command documentation under apoptosis)
- Tie-impasse substates now augmented with *non-numeric* and *non-numeric-count* attributes, analogous to *item* and *item-count* attributes.
- Soar will no longer automatically change the indifference selection policy to epsilon-greedy when turning reinforcement learning on

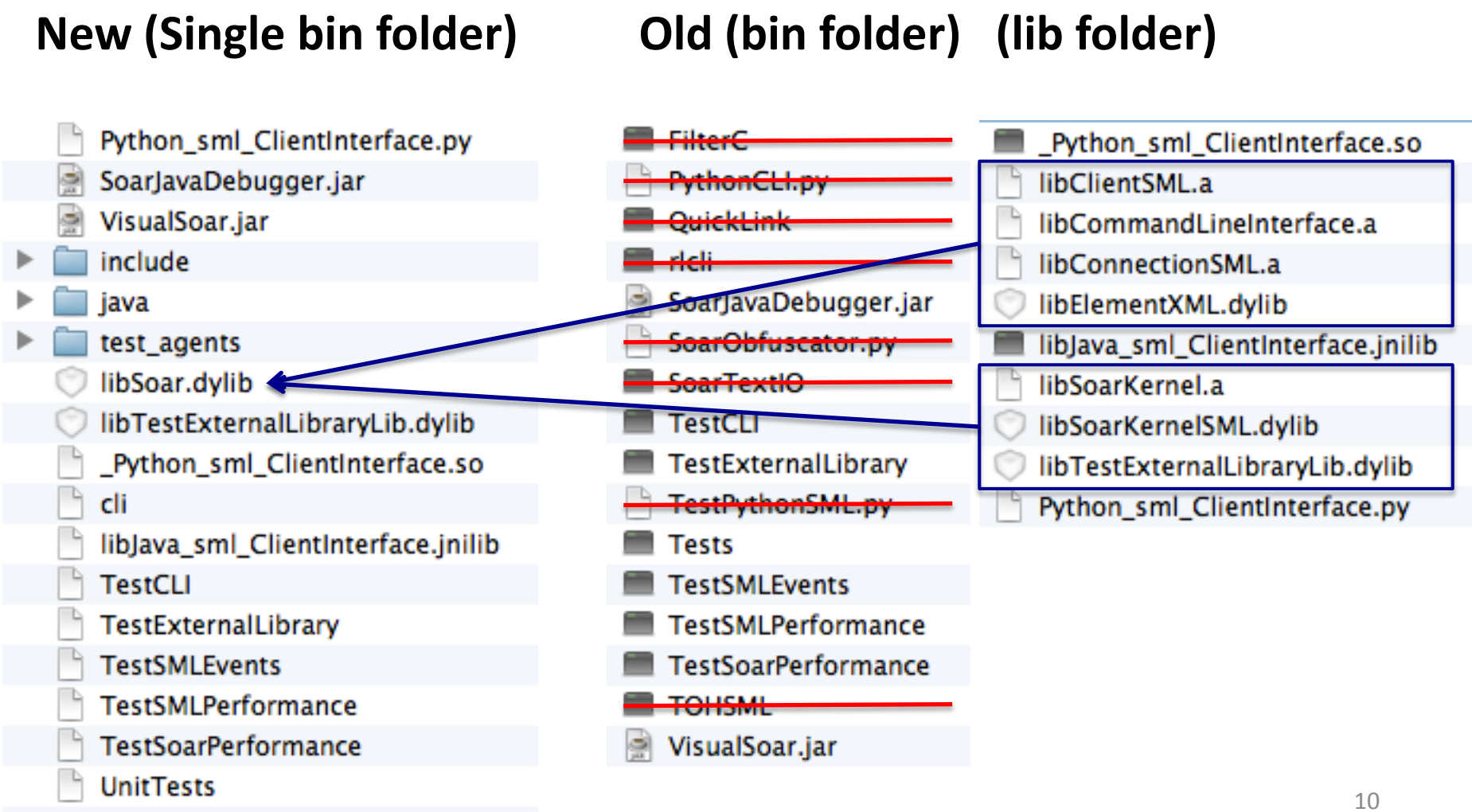
New Project Organization

- SoarSuite streamlined (8mb)
  - Contains the Soar kernel, Java debugger, VisualSoar, the Soar manual, default rules, unit tests and Soar header files
  - Other components moved to separate downloads
- The Soar Tutorial targeted towards people learning Soar for the first time.
  - In addition to SoarSuite contents, contains the Soar Tutorial, Eaters, TankSoar, and all the introductory agents discussed in the tutorial and John's new book

What's in Soar Suite 9.3.2



What's in Soar Suite 9.3.2

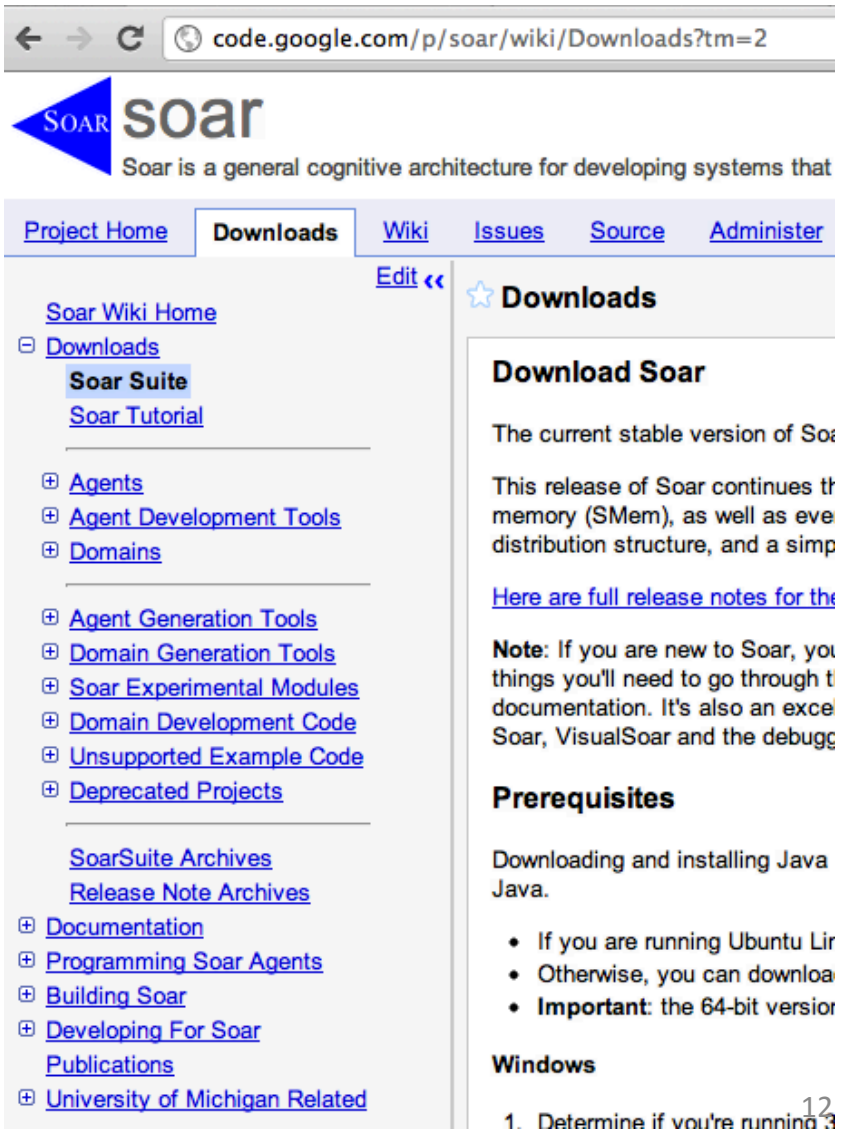


Improved Soar Web Pages and Wiki

- Nearly everything has stand-alone versions to download. Each project has its own web page.
- Every agent in every domain now has its own web page with thorough descriptions
- We encourage you to submit your own agents, tools and domains for inclusion. We want this to very much be a community area.

Improved Soar Web Pages and Wiki

- Agents
  - All agents from every domain
- Agent Development Tools
  - Soar Editor, VisualSoar, Vim plugin, UltraEdit config, Epmem-size, QuickLink, State2Dot
- Domains (8 project)
  - Dice, Eaters, TankSoar, InfiniteMario, RoomsWorld, SoarQnA, SoarTextIO, WordNet
- Generating Tools
  - PDDL, GGP
- Unsupported & Deprecated Projects
- Documentation
  - tutorial, manual



Improved Soar Web Pages and Wiki

- Contacts, alumnae, Soar research groups, publications all updated and dead links removed
  - Please contact us if you want you or your work added to any of these pages.
- Much obsolete documentation removed from Wiki
- Many new documents added to wiki and re-organized to make things easier to find

Coal

- Many older projects are no longer actively supported.
  - All are now under either the example development code or unsupported category.
- Some minor SML API changes

Nuggets

- Compilation is more “out-of-the-box”
- Spring cleaning. Lots of obsolete things removed. People should be much more confident the information they are reading is current.
- It should be much easier for people learning Soar to browse what people have put together in the past.
- Consistently formatted and comprehensive pages for all projects should promote code re-use. People can see everything available quickly and how to get it.