Removing gSKI from the Kernel

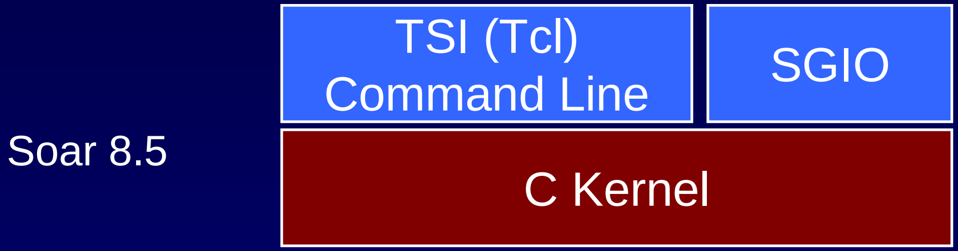
Douglas Pearson
Bob Marinier

Soar 27: May 24th, 2007

doug@threepenny.net

ThreePenny SOFTWARE

A Little Kernel History



Perceived problems:

- Building external interfaces to Soar was hard
- Kernel interface was not cleanly defined
- Hard to extend tool set (Tcl only language and tied into kernel)

Different paths to a solution

- Scott Wallace – new C++ interface to the kernel
- Soar Tech -- gSKI, a different C++ interface to the kernel
- U of M/3p -- SML, XML-based remote interface to the kernel

What should Soar 8.6.0 be?

ThreePenny SOFTWARE

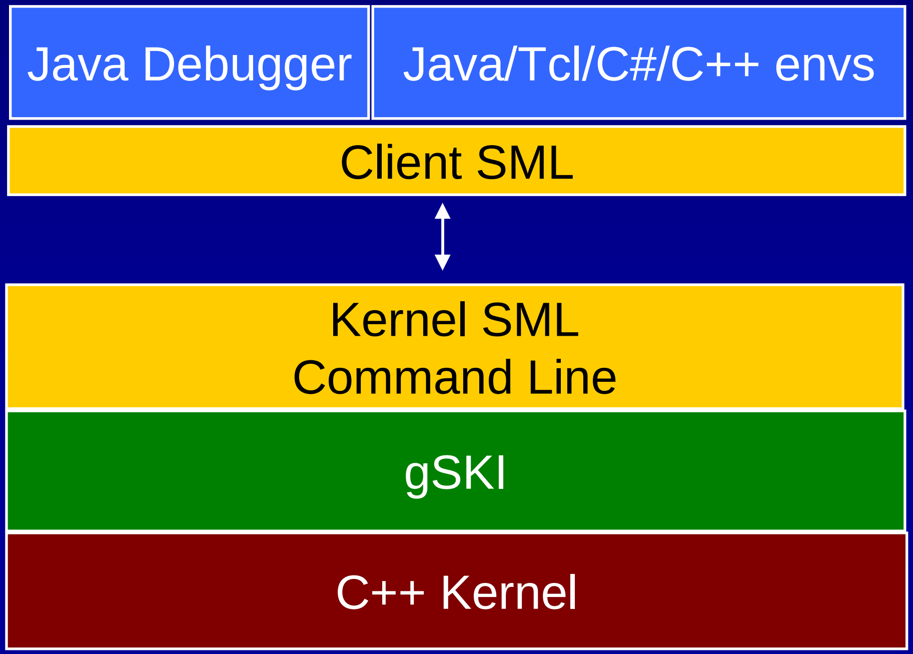
A Little More Kernel History

Major concern that 8.6 would fracture Soar

- University-Soar
  - Would use SML for remoting/external interfaces
  - Wouldn’t benefit from object-based C++ interface to kernel
- SoarTech-Soar
  - Would use gSKI for remoting/external interfaces

SoarTech generously contributed gSKI to the community leading to 8.6.0

Soar 8.6.0



Why removing gSKI now?

- Intended to make external interfacing easier
  - But community interfaces through SML
  - SoarTech interfaces through ATE (alternative XML interface)

- Added to code complexity for kernel developers

– Kernel	~42,000 LOC	
– KernelSML	~20,000 LOC	
– gSKI	~42,000 LOC	(incomplete implementation)

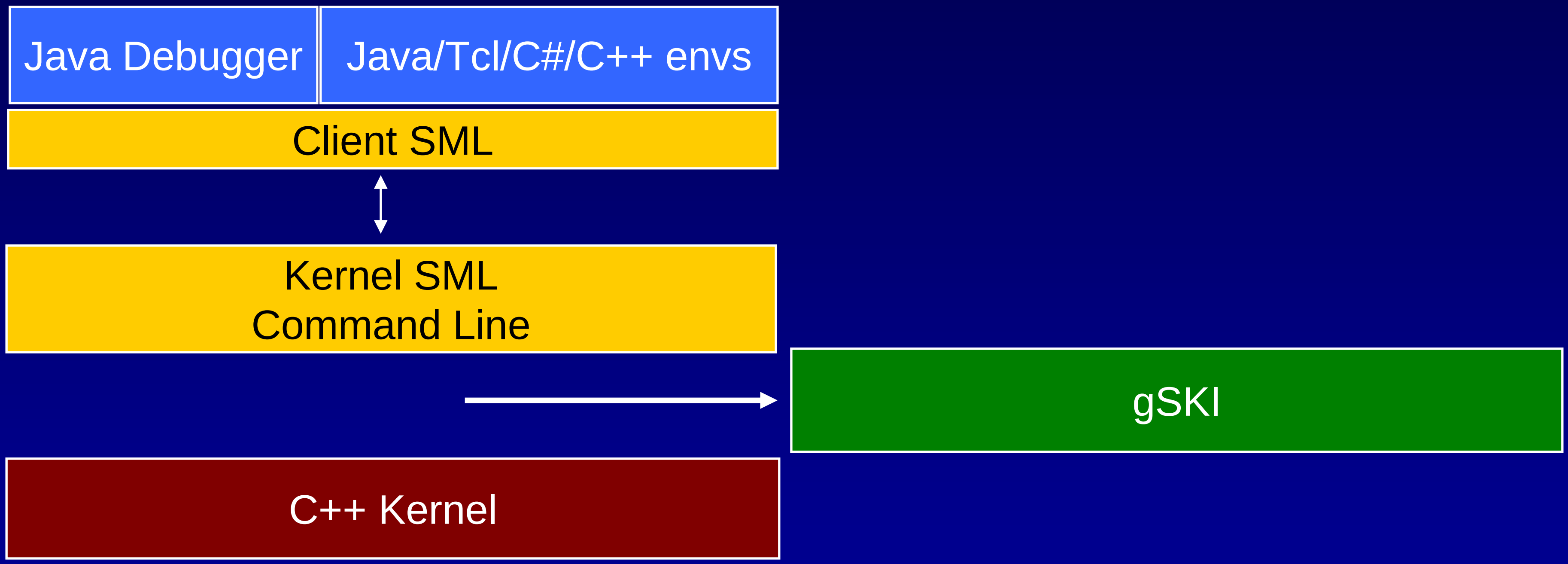
- Abstraction presented by gSKI different from that of kernel
  - Different assumptions about I/O
  - Problems with memory management

- Extending the kernel requires adding code to gSKI layer as well

ThreePenny SOFTWARE

How removing gSKI?

- Sliding it out of the kernel stack



Maintain a working system at all steps along the way as remove gSKI pieces.

Removing vertebrae from a patient without putting them to sleep.

ThreePenny SOFTWARE

Expected Impacts

- API changes
  - Goal is to have no impact to clients
  - No change to debugger, environments
  - A few events removed that were never implemented (if you're using them you're confused as they do nothing)
  - Possible that some event ordering could change
- Kernel developers
  - Very little change inside the kernel code
  - Mostly removal of gSKI specific events, replaced with kernel callbacks
  - Should have very little impact on people adding on to the kernel
- Performance changes
  - Should be negligible and what there are should be positive (less code)
- Major impact is in KernelSML code, but few folks work in there

Nuggets and Coal

- Nuggets
  - Client/Kernel clean separation very helpful here
    - No way for a client to call to the kernel directly
  - Lots of test tools written already for 8.6.0
    - Bob running all tests after each round of changes
  - Most of conversion complete and incremental approach worked
    - Command line interface
    - Events (includes a lot of functionality, e.g. external rhs functions etc.)
    - Output
    - Input
- Coal
  - Should have figured this out earlier
  - Could be introducing subtle bugs not revealed by tests
  - As do final removals will probably find some pieces not converted