gSKI

generic Soar Kernel Interface

Soar Technology

What is gSKI?

- Advanced All Purpose Interface to the Soar Kernel and Related Tools
- Yet Another Kernel Interface

Why gSKI?

- Interfacing the Soar Kernel to new applications takes too long.
- We need an interface that can talk to, not only a running simulation, but to tools and other components as well.

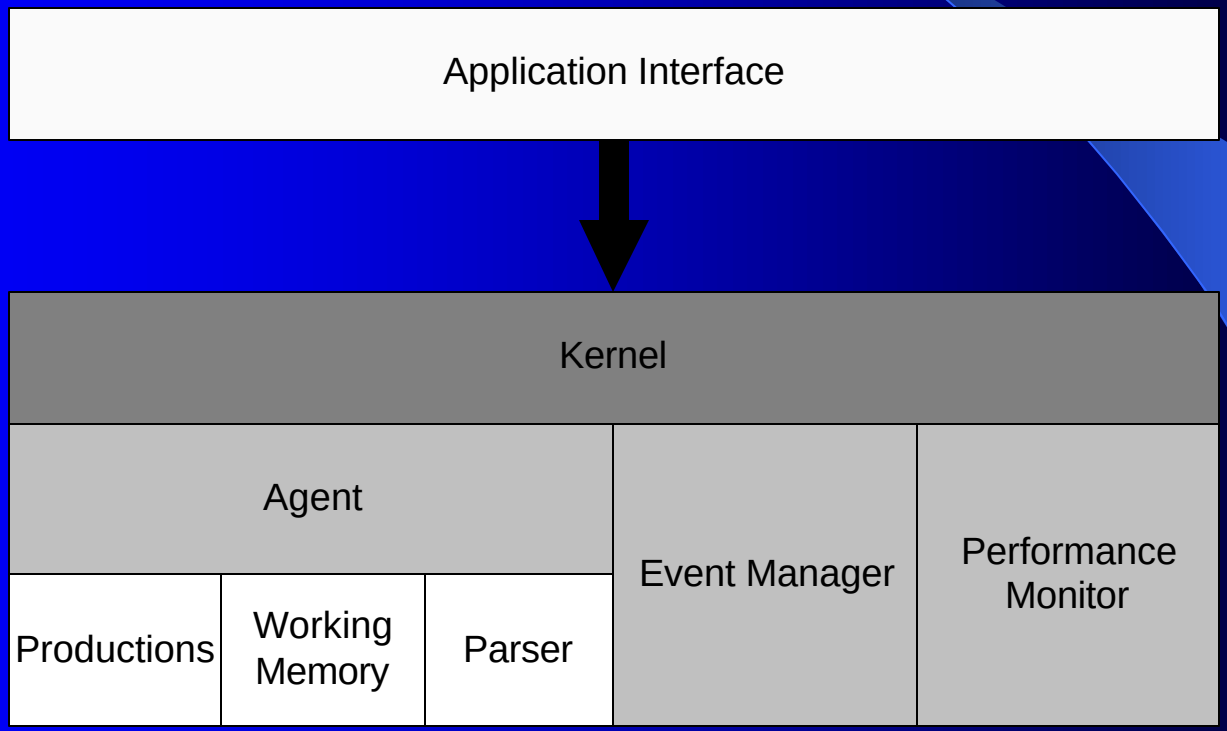
Why not an existing API?

- The lowest level API (8.3)
  - Very low level
  - Messy interface (soarkernel.h)
  - Requires knowledge of kernel internals
  - Easy to shoot yourself in the foot
  - Exposes Kernel internals
- The ‘C’ API
  - Kernel wrapper for TCL
  - Doesn’t provide the full interface to the kernel
  - Still too low a level
  - Insufficient documentation
- SGIO
  - Shallow wrapper around the ‘C’ API
  - Even more limited that the ‘C’ API
  - Non user-centric design

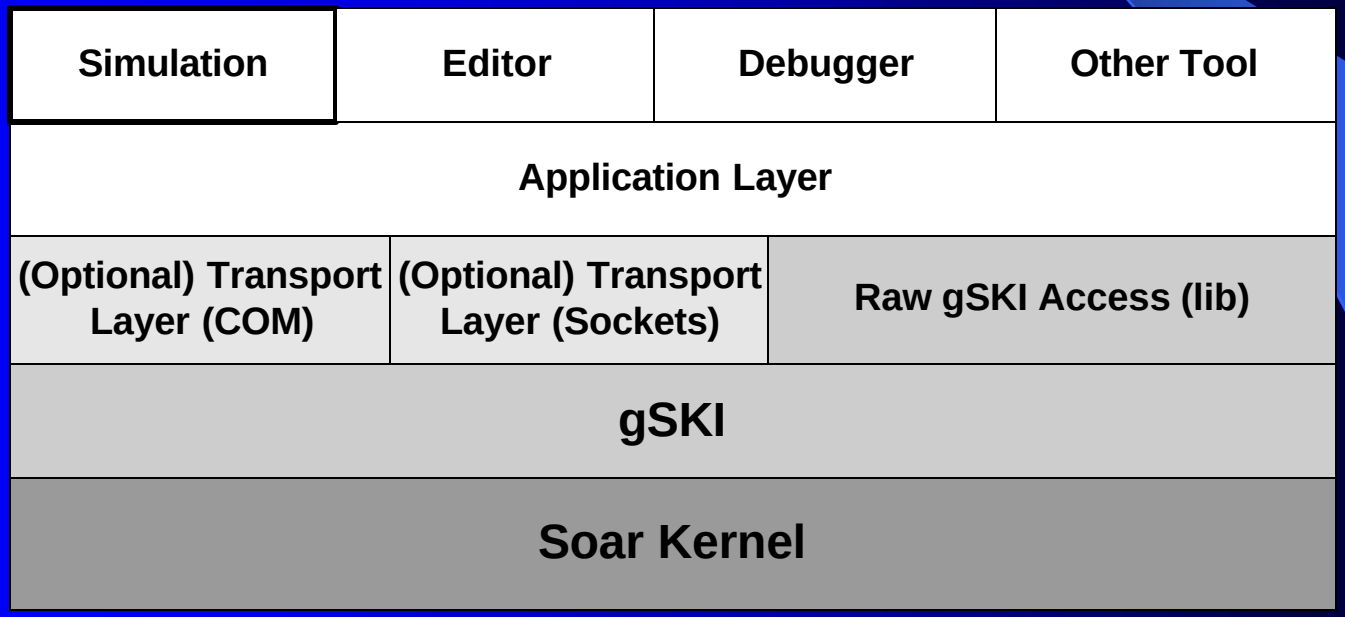
What will gSKI do for me?

- Provide a robust, uniform, well abstracted interface to the Soar kernel
- Require only a conceptual understanding of the kernel internals
- Hide internal kernel data structures
- Provide efficient access to the Kernel
- Backward compatible with existing .soar files through a new TCL/gSKI interface
- Provide good documentation for kernel interfacing

Primary Interfaces

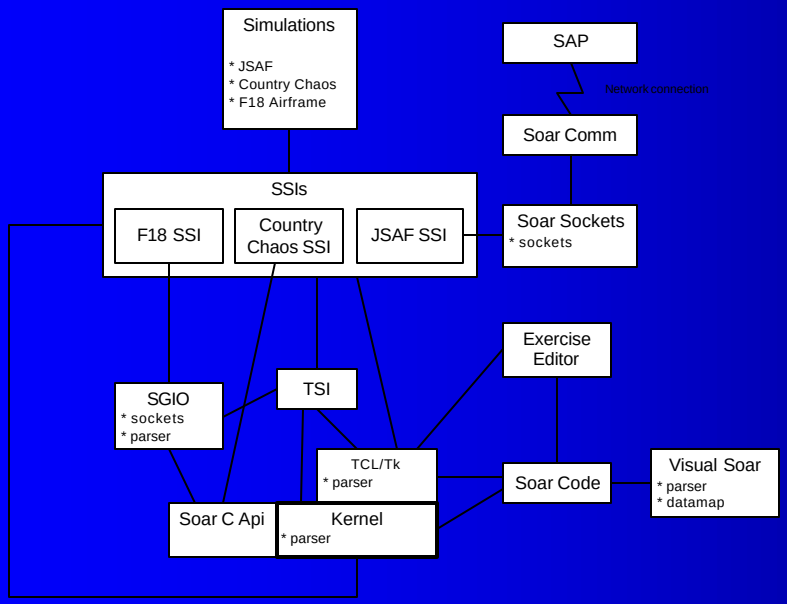


System Structure

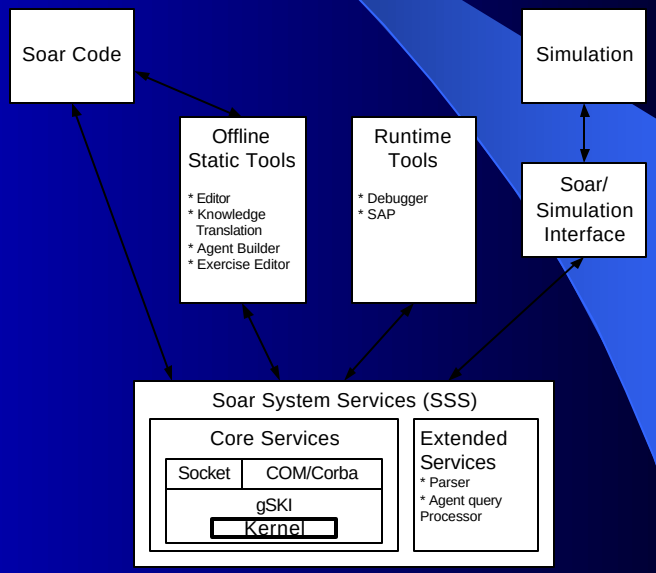


The Road to gSKI

Before



After



What is a ".soar" file?

- Currently a ".soar" file is a TCL file with some Soar specific functions built in.
- gSKI is going to divorce TCL from the Soar Kernel and provide a mechanism for processing ".soar" files without using TCL at all.

Conclusions

- We need to interface to the Soar Kernel quickly and efficiently.
- There is currently not an interface the meets our requirements.
- We are designing gSKI to be our new standard interface to the Soar Kernel.

Credits

- Produced By
  - Soar Technology

- Team Lead (gSKI Instructor)
  - Al Wallace
- Design and Implementation
  - Scott Colcord
  - Jacob Crossman
  - Rich Fredricksen
  - Frank Koss
  - Jens Wessling