Knowbot® Programming

Jeremy Hylton
Corporation for National Research Initiatives

joint work with Roger Masse, Fred Drake,
Ken Manheimer, Guido van Rossum,
and Barry Warsaw

Motivations & Biases

- Knowbot Programs (KPs)
  - mobile
  - agents, loosely defined
  - focus on systems issues
- Why mobile programs?
  - move programs closer to resources
  - autonomous application components
  - applications: couriers, indexing, search & retrieval

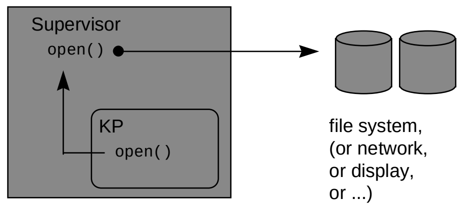
KOS Architecture

- Knowbot Operating System (KOS)
  - runtime environment
  - runs user KPs & KOS extensions
  - simple programming environment
  - extensible, e.g. multiple languages
- System infrastructure
  - connectors: object RPC
  - migration: mobility & state management
  - security: protect KOS from KP
  - also: namespace, asynchronous events

How It Works

- Prototype system
  - KPs written in Python
  - uses ILU for communication
  - KOS implemented in Python
- Knowbot operating environment
  - KOS run on servers
  - users submit KPs
  - KOS reports progress to user

Security



- Restrict KP's access to host system
  - Safe language, no pointers
  - restricted execution environment (padded cell)
  - trusted supervisor mediates access to resources
- Lots more to do

9/26/96 KOS Overview 5

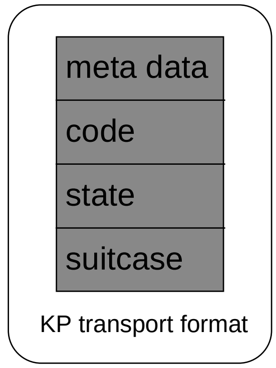
Connectors

- Thin layer over object RPC
  - lookup by name, type
  - restricts KP access to RPC
  - access control
  - can be moved

9/26/96 KOS Overview 6

Migration

- Getting from here to there
  - `migrate()` or `clone()`
  - entry point: `__main__()`
  - carries pickled state and suitcase
  - metadata:
    - originator
    - programming language
    - completion action



Where to Go from Here

- Applications
  - distributed indexing
  - search & retrieval
  - couriers (intellectual property)
- Future work
  - Security
  - Multiple KP languages
  - Resource allocation
  - Instrumentation, study performance