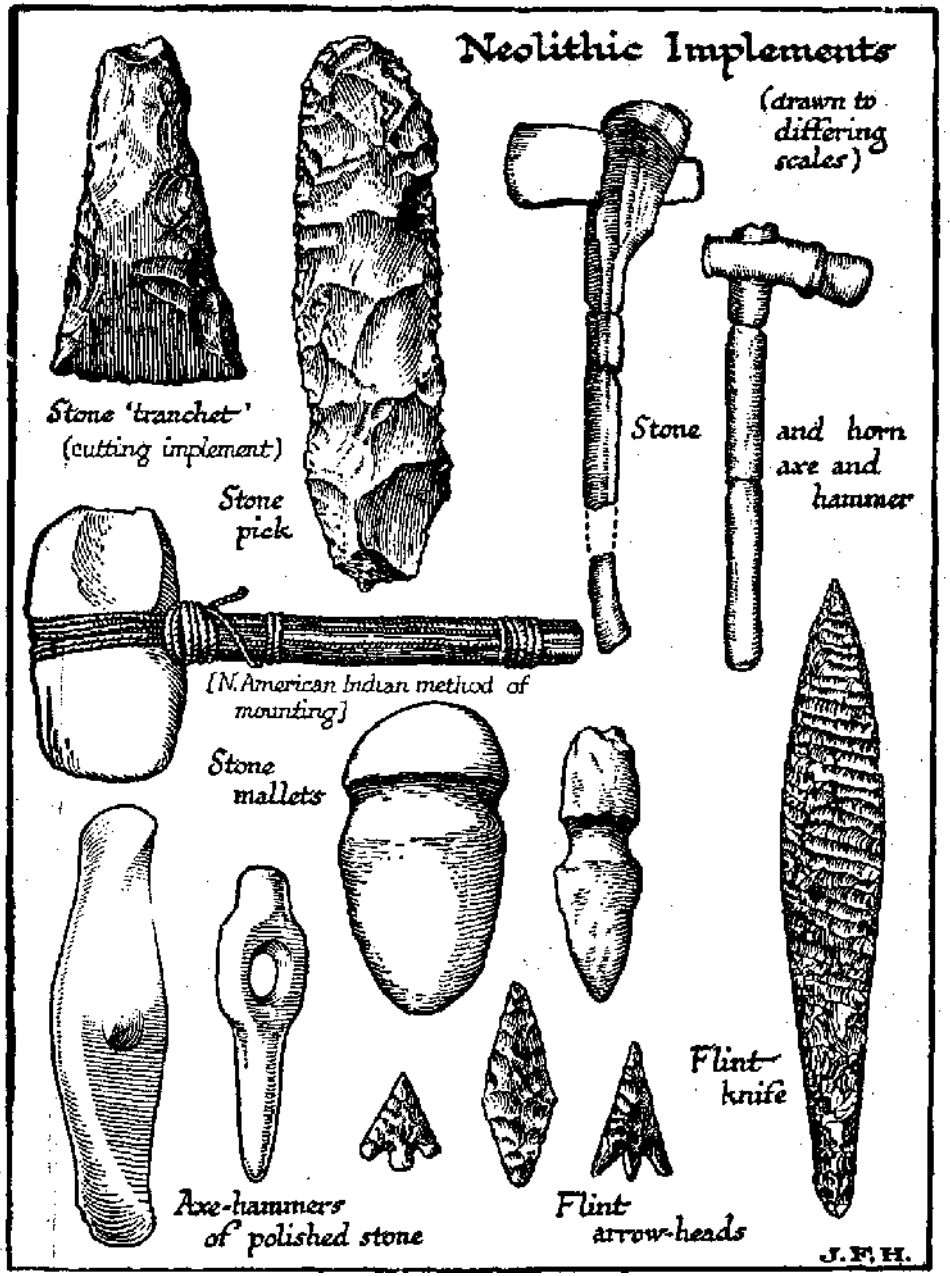
TOOL


Neolithic Implements
(drawn to differing scales)
Stone 'tranchet' (cutting implement)
Stone pick
Stone and horn axe and hammer
[N.American Indian method of mounting]
Stone mallets
Flint knife
Axe-hammers of polished stone
Flint arrow-heads
J.F.H.

New Modular Library

- Specially Geared for TclOO Development
- Stresses the development of Frameworks rather than discrete packages
- An effort to take the useful bits out of Tao for produce a standard set of design patterns

WORK IN PROGRESS

Major Components

- TOOL Core Framework
- Sherpa
- TOOL Modules

Dicttools

- dict rmerge
  - Recursive Merge
- dict is_dict
  - Boolean test for if a value is a valid dict (without shimmering it into a list)
- dict print
  - Pretty indented output
  - Works recursively
- dict getnull
  - Return an empty list if a key isn't present instead of throwing an error

localtool

- OO Toolkit built on TOOL Core
- Supersedes Nettool
- Provides local platform introspection
  - Canonical TEAPOT platform code
  - OS Specific application storage location
  - Tools for finding external executables
  - Wrappers for common utilities that require different OS or platform specific calls
  - Network interface introspection
  - Access to unique hardware keys

TOOL Core Framework

- Supersedes the oo::utils package in Tcllib
- Distributed with Tcllib
  - In the “odie” feature branch
- Distributed as its own fossil repo:
  - http://fossil.etoyoc.com/fossil/tool
- Distributed with sherpa

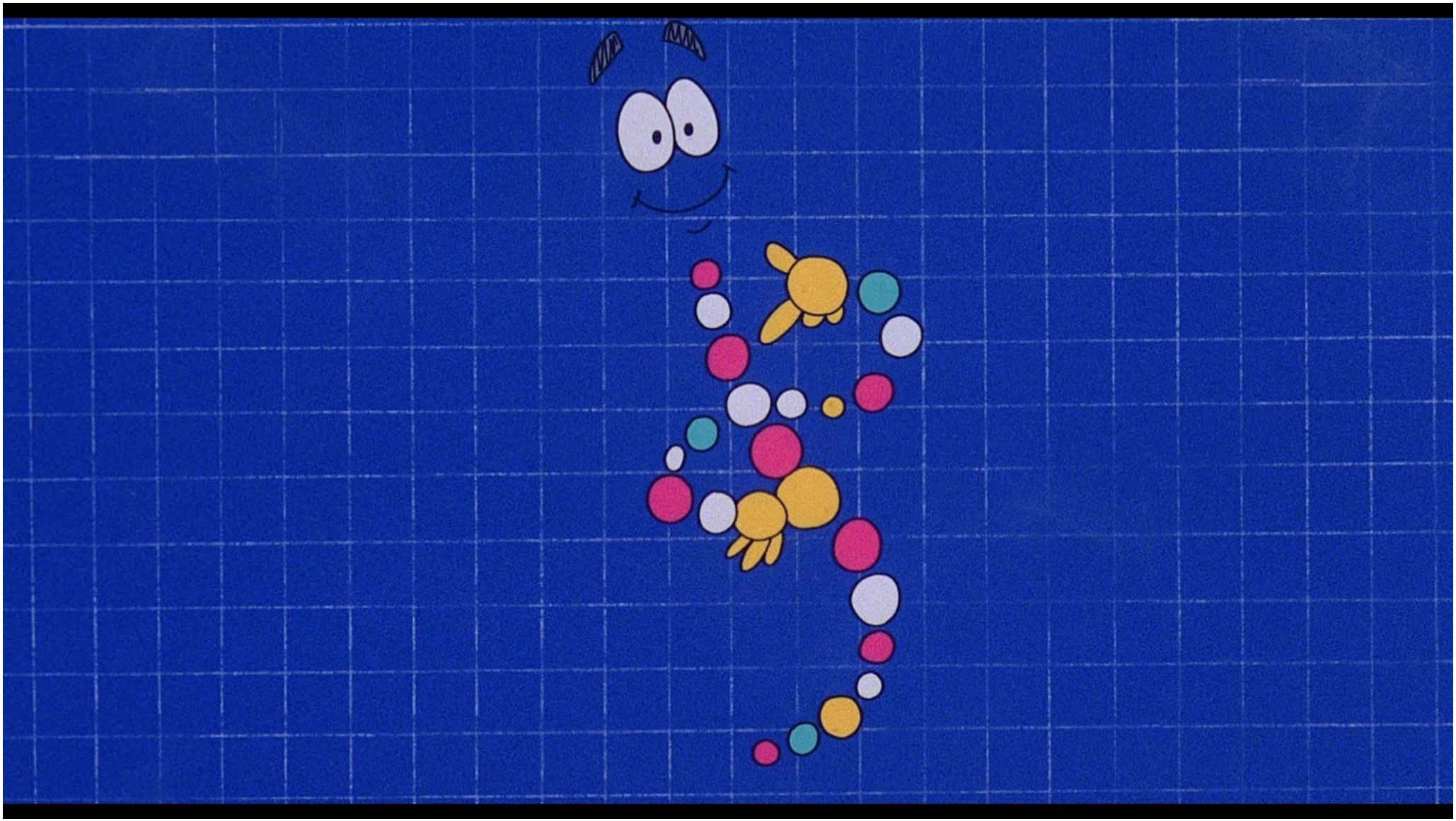
So Why TOOL?

- Tcl Integration today is akin to “Code Recycling”
- Developer pulls parts from various sources
- Developer hacks and grafts the parts together


2011

Code Inheritance

- Tcl Integration in OO is more like genetic splicing
- Components combine holistically
- Behaviors combine complexly and occasionally in non-intuitive ways



OO Framework Requirements

- Documentation needs to track with versions
  - Goal is to use markdown which is checked in alongside of the code
- Ability to work directly from fossil repository
-

Sherpa

- Yes… THAT sherpa
- Has been split from the “odie” project
- Is distributed as a separate fossil repository:
  - http://fossil.etoyoc.com/fossil/sherpa

Sherpa

- Designed to download and compile extensions
- Is aware of, and can exploit, the Teapot instead
- Can
  - install a project locally
  - build a teapot module
  - Inject either a local project or a teapot module into a VFS

Still a work in progress

- Thank you Brad for bravely trying it out last night…
- Emphasis on “tried”

Sherpa Components

- Command Line Interface
- Interactive Shell
- Recipe Builder
- Developer Convenience Shell