

Medley Interlisp

Features Demos

A Residential Programming Environment supports Incremental Development

A residential programming environment is one in which you define, edit, debug in the live environment; write out to files to save for future session. The system tracks what changed. Multiple features work together, to act as a "programmer's assistant":

DWIM - Do What I Mean, Spelling Correction: DWIM is more than just spelling correction; it's more an attitude -- not to signal errors if what the user meant is obvious and correctable.

```
(DEFINEQ (FOO1 (X) (PLUSS X X]      <-- PLUS misspelled.
                                           < Note []

superparenthesis.
PP FOO1
(FOO1 6]
Y
PP FOO1
```

History and UNDO: Interlisp was the first known system. DWIM changes are undoable.

```
??
UNDO <dwim event>
PP* FOO1
```

- The file manager tracks what's been defined or changed. The file "coms" are like a manifest -determines what goes where.

```
FILES?)
```

Y FILEB
 (CLEANUP)
 SEE FILEB

- *Helpsys and DInfo (also a first) interactive documentation*

MAN INFILE
 MAN CL:WITH-OPEN-FILE
 MAN CLHS.OPENER
 <right click DInfo>

- *Masterscope (interactive cross reference) has an extensible english-like language for querying*

LOAD(HELPSYS PROP)
 . ANALYZE ON HELPSYS
 . SHOW WHERE ANY CALLS CLHS.OPENER
 USE EDIT FOR SHOW
 . SHOW PATHS FROM HELPSYS

Common Lisp and Interlisp Integration

an Inter-medley of Uncommon Lisps

- freely intermix CL and IL function, macro, variable definitions
- All datatypes are common: lists, NIL, symbols, arrays, strings, structures, numbers
- many functions and special forms are the same: CAR, COND, ED
- Some are slightly different: CL:EVAL vs. IL:EVAL, CL:LAMBDA vs. IL:LAMBDA, ...
- Extend CL to include IL and IL to include CL
- Common Lisp definitions and declarations managed by Interlisp environment

Demo:

```
. SHOW WHERE ANY ON HELPSYS CALLS CL:WHEN  
USE EDIT FOR SHOW
```

Structure editors

`EDITMODE(TELETYPE)` original structure editor: powerful, programmable, short commands (BO 3) P (SW 1 2). Useful for editing huge list structures or Lisp functions.

`EDITMODE(DEDIT)` (Load it to try). Like the TTY editor with menu and showing the results in a window.

`EDITMODE(SEDIT:SEDIT)` Default editor. SEdit maintains illusion that you are editing structure. Parentheses are balanced at all times, but you can just type and backspace. Keyboard controls and attached menu.

Common to all: user never counts parentheses, modifies whitespace, or needs to fiddle with line breaks or indentation

The Virtual Machine and OS

- D-machines had microcode to interpret a bytecoded instruction set. Subsequently, the microcode was reimplemented in C (named "maiko").
- "Sysouts" (memory images) can be moved from machine to machine; only maiko needs to have been compiled for each OS and chip architecture.
- **Medley is small** (relatively speaking). Bytecodes are compact. Medley online uses 16MB/user (64MB max). Installed, 256mb max. Time to make new image 15 seconds; restart a saved image "in the blink of an eye".
- Interlisp-D was the **whole operating system**: scheduler, window manager, network, drivers. Now can rely on host OS for device drivers

GIT: A Repository for Medley Definitions

Conventionally, GIT tracks and compares files

- Change detection and presentation based on line-editing semantics: Mismatching character sequence => significant difference

Medley's source files: external archives for structured definitions with metadata

- Saved and loaded, maybe printed, but never edited
- A given definition can be represented in different (but semantically equivalent) line and character sequences

Medley interface to GIT: Definition-based change tracking

P(ull)R(equest)C(ompare) command (demo)

- Retrieve changed files from GIT, find/parse PR vs master alternative definitions, compare as Lisp-structure differences

Idea: load and manage definitions @ commit level granularity

- If function FOO is included in commits C and D, definitions FOO;C and FOO;D are co-resident.
-

LispUsers, Library, Internal

(Favorite LispUsers)

Some other development tools: SPY, File Browser

Revive Applications using Interlisp

Notecards - early HyperText

Rooms - Screen / Desktop management

LOOPS - Lisp Object Oriented Programming System (not CLOS)

Truckin' - LOOPS game for teaching "Knowledge Representation"

LFG - Lexical Functional Grammar

.... and others