

# the Common Lisp Interface Manager (CLIM)

Robert Strandh  
Université Bordeaux 1

currently on sabbatical at the  
University of Auckland CS department

# Contents of talk

- CLIM
  - flow-driven as opposed to event-driven
  - layered collection of protocols
  - platform independent (different backends)
- McCLIM, a free implementation of CLIM
  - how the project got started
  - where we are now
  - future development
- Demos

# CLIM

CLIM is a *specification* that is publicly available.

Commercial implementations exist for Allegro, LispWorks, MCL, Genera.

It is essentially an abstraction of the Lisp-machine presentation facility.

## CLIM is flow-driven

Most GUI tool kits require event-driven programming

You can program this way with CLIM as well, but CLIM contains functions to turn events into streams.

You can therefore use your normal programming style using the recursion stack and the value of your program counter as usual.

# Layered collection of protocols

CLIM is a large layered collection of protocols

Allows advanced users to intervene at every level by modifying behavior or adding new functionality.

At the lowest level, there are relations between windows (called sheets) and at the highest level, there is functionality for graph formating, table formating, structuring of applications, etc.

## Many useful built-in features

CLIM is similar in spirit to Common Lisp itself: have a relatively small, sound core plus a very large collection of useful built-in features on top of the core.

Examples:

- output recording
- graph formatting and table formatting
- predefined presentation types for many CL types
- incremental redisplay
- automatic completion

# Platform independence

Multiple backends possible:

- 'native' CLIM look
- platform-specific backend using native widgets

## Key feature: presentation types

Presentation types are parameterized types that form a type lattice, parallel to that of Common Lisp.

Using `present` to display an object creates a *presentation* that includes the graphic display, the presentation type, and the underlying object.



# McCLIM

In 1999, I needed a toolkit for my score editor (Gsharp) which is written in Common Lisp.

After some research, it seemed CLIM was the obvious choice, but no free implementation was available, though some code from Mike McDonald and Gilbert Baumann existed.

In 2000, I therefore started developing what later became McCLIM.

## Current state of McCLIM

- 100kLOC (35k core, 20k backends, 45k applications and tools) of Common Lisp
- most CLIM features are implemented
- a few performance problems (incremental redisplay)
- some conformance issues that create problems for advanced applications (like Gsharp)

## Existing backends

- CLX (same level as Xlib)
- OpenGL (still incomplete)
- Cairo
- Beagle
- PostScript
- (HTML) unknown status

# Contributors

- Mike McDonald, Gilbert Baumann,
- Alexy Djneka, Tim Moore
- Robert Goldman, Iban Hatchondo, Julien Boninfante,
- Lionel Salabartan, Arthur Lemmens, Arnaud Rouanet
- Andy Hefner, Christophe Rhodes,
- Andreas Fuchs, Max-Gerd Retzlaff
- and probably many more

## What is being worked on

- the Cairo backend
- applications (Gsharp, Climacs, etc)
- regression test framework

but mostly, McCLIM is currently without a “project leader” .

## Wish list, future projects

- replacing Goatee with Climacs
- faster incremental redisplay
- more modern “look”
- more backends (Gtk, Windows, etc)

# Demos

- Gsharp
- Climacs
- Accounting program

Questions?