

McCLIM Demonstration

Daniel Kochmański
TurtleWare – Daniel Kochmański
Przemyśl, Poland
daniel@turtleware.eu

ABSTRACT

We describe what is a *Common Lisp Interface Manager*[2] implementation called McCLIM[7]. In particular, we describe recent improvements of the code base. We illustrate McCLIM and recent development by developing a demo application “Clamber”, which is a book collection management tool, which was created in purpose of explaining *CLIM* concepts in form of a tutorial.

CCS Concepts

•Software and its engineering → Integrated and visual development environments;

Keywords

Common Lisp, graphic user interfaces

1. INTRODUCTION

The CLIM specification[3] is large and requires some initial work from the programmer to start writing programs using CLIM. Needless to say, the implementation of such specification is a non-trivial undertaking and indeed, McCLIM development consumed many man-hours of a bright developers through the first decade of XXI century. In the second decade development pace slowed down a little to get on track again during the last few years, thanks to individuals like Robert Strandh (current project leader), Alessandro Serra and others.

The *CLIM 2.0* specification was released in 1993. It is meant to be portable across Common Lisp[1] implementations and operating systems by mapping itself to the target window system (like X11[5], GTK[4] or Windows Forms[6]) through backends.

The key feature of the standard is providing a convenient object-oriented abstraction over the interface presented to the user (without compromising programmer’s *ability* to change low-level details). With these things in mind it is a specification which implements the model-view-presenter archi-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
ELS '17, April 3 – 6 2017, Brussels, Belgium Copyright is held by the owner/author(s).

tectural pattern in a consistent way while also providing defaults and the ability to customize its behavior.

McCLIM is a free open source implementation of CLIM II specification with extensions proposed by Franz Inc. in the *CLIM 2 User Guide, version 2.2.2*. As of 2017, McCLIM (and recently opensourced *clim2*¹) is the only available native graphic user interface toolkit available to the Common Lisp ecosystem. Other solutions are based on foreign tools (LTK², CommonQt³ or EQL⁵) or are commercial (Common Graphics⁵, CAPI[?]). Another frequently used approach is creating web applications with frameworks.

A few applications and libraries written in McCLIM are shipped with McCLIM code repository:

- Listener

The McCLIM Listener provides an interactive toplevel with full access to the graphical capabilities of CLIM and a set of built-in commands intended to be useful for Lisp development and experimentation.

- Inspector

Clouseau allows inspecting arbitrary Common Lisp objects and provides a disassembler wrapper.

- Debugger

Graphical debugger using portable debugger interface developed for the Slime project (Swank). Graphical look and feel is inspired by Slime.

- ESA[8]

A CLIM Library for Writing Emacs-Style Applications.

2. RECENT DEVELOPMENT

During the last year we have replaced various functionalities with third party systems. The *Bordeaux Threads* library is responsible for threading, *OptiCL* for handling raster images etc. We have also refreshed the McCLIM website to provide up-to-date information. Some work has been done to assure its portability across implementations. As of 2017-01-25 it is known to work on *CCL*, *ECL* and *SBCL*.

¹<https://github.com/franzinc/clim2>

²<http://www.peter-herth.de/ltk/>

³<https://common-lisp.net/project/commonqt/>

⁴<https://gitlab.com/eql/EQL5>

⁵<http://franz.com/support/documentation/8.2/doc/cgide.htm>

After that, we have started a crowd funding campaign which allowed us to finance bounties and some constant development work (40 hours per month). It was a huge success which assured us, that people are lively interested in seeing *McCLIM* working. We have gathered very valuable feedback which helped us to plan a roadmap⁶ for the development. We regularly present an iteration summary to the community. We have noticed a development steady progress and growing interest in using *McCLIM* by the Common Lisp developers.

A lot of effort is put into *McCLIM* internals to assure quality and ease of use, and to provide learning material for new users. One of the most discouraging parts of using CLIM is the software complexity with little material that helps to understand how to use it.

3. DEMONSTRATION

Key CLIM components are:

- Application Frames

Application frame is a class abstracting one application entity. It is composed of the application internal state, its panes, layouts and commands associated with it.

- Panes, layouts and views

Panes are more widely known as widgets. Layout is a special kind of pane responsible for the application frame screen composition. Views may be used to customize the rendering according to the particular needs (textual view, gadget view, list view, grid view etc).

- Drawing primitives and formatted output

For custom output, the programmer disposes of a rich set of drawing functions which may be used on a sheet. Also, output may be arranged in tables, graphs and lists which helps to lay things out on a sheet when writing custom widgets.

- Presentation types

This feature allows the programmer to display Lisp data on the screen. The displayed data is associated with a reference to a particular (presentation) type.

- Commands

Commands are functions provided by the application to achieve its design goals. Command arguments may be gathered by various means available to the user – through a gesture, by selecting a command in a menu or simply typed in the *Listener*, which allows a clear separation of application logic and its look.

4. CONCLUSIONS AND FUTURE WORK

We have demonstrated that *McCLIM* is an interesting piece of software which allows the creation of customizable user interfaces. Convenient abstractions enable the programmer to think in high-level terms which are well mapped to the underlying functionality.

Recent spawn of interest is a proof that free CLIM implementation is something demanded, and we plan to improve it to the point where it is stable enough for production use.

⁶<https://common-lisp.net/project/mcclim/involve>

One of the primary goals is providing a good documentation and learning materials for newcomers, since the specification is vast (and not necessarily consistent).

5. ACKNOWLEDGMENTS

We would like to thank all the contributors, supporters and programmers using *McCLIM* for all the valuable feedback, code improvement, financial support and encouraging words.

I personally want to thank professor Robert Strandh for his constant patience and mentorship which made this work possible and entertaining.

6. REFERENCES

- [1] *INCITS 226-1994[S2008] Information Technology, Programming Language, Common Lisp*. American National Standards Institute, 1994.
- [2] S. McKay. *Clim: The common lisp interface manager*. *Commun. ACM*, 34(9):58–59, Sept. 1991.
- [3] S. McKay and W. York. *CLIM 2.0 Specification*. <http://bauhh.dyndns.org:8000/clim-spec/> [access 2017-01-30], 1993.
- [4] H. Pennington. *GTK+/Gnome Application Development*. New Riders Publishing, Thousand Oaks, CA, USA, 1999.
- [5] R. W. Scheifler and J. Gettys. The x window system. *ACM Trans. Graph.*, 5(2):79–109, Apr. 1986.
- [6] C. Sells and M. Weinhardt. *Windows Forms 2.0 Programming (2Nd Edition) (Microsoft Net Development Series)*. Addison-Wesley Professional, 2006.
- [7] R. Strandh and T. Moore. A free implementation of clim. In *Proceedings of the International Lisp Conference, ILC 2002*, October 2002.
- [8] R. Strandh, D. Murray, T. Henriksen, and C. Rhodes. *ESA: A CLIM Library for Writing Emacs-Style Applications*. In *Proceedings of the 2007 International Lisp Conference, ILC '07*, pages 24:1–24:10, New York, NY, USA, 2009. ACM.