Empirical Modelling and the use of LISP macros

Jeffrey Downie, Student ID: 1003773

November 20, 2013

Abstract

My project proposal is inspired from the prevalent use of macros within current teaching models in Empirical Modelling. Within many of these models we have lines upon lines of code all of the form "*board[1][1] = "black"; board[1][2] = "white";...*," and so on for 64 lines. From the sheer prevalence of this sort of macro-assisted coding, it appears to be worth investigating as to whether stronger macro integration into Empirical Modelling tools would be worth pursuing. I therefore wish to attempt to use the inarguable king of macro coding, LISP, with dependencies from Empirical Modelling.

Other aspects of LISP also lend themselves to Empirical Modelling. LISP lends itself well to (and is commonly used as such) adding to code via an interpreter on-the-fly, as is currently being done in Tkeden, where you enter each code block/function (re-)definition in turn, and can see the results immediately, without having to re-set the state of the program. This allows for quicker response, and hence quicker learning from the models, a key objective of the Empirical Modelling project.

This is not, however, a complete match made in heaven. LISP is perhaps the ideal in functional form, whereas Empirical Modelling is heavily reliant on remembering states and procedural code. A critical analysis on this integration of opposite styles of programming must therefore be the heart of this paper.

1 Modelling Study.

Before even creating the models to be used in this critique, we must first have the language available to us to write them this way. Similarly to how LISP uses *(set x '(...))* as its equivalent of *x = ...;*, I will add *(defdep x '(...))* as an equivalent of *x is ...;* to LISP. In order to achieve this, I will write a (mostly) complete LISP interpreter in order to add the use of dependencies into the language. The complete re-write of an interpreter appears unfortunately necessary in order to re-define how *set* works so that variables will update correctly.

The models that shall be made will be attempts to clone previous projects such as the pjawns model in order to fairly compare the two approaches to creating the models. Other such models could be taken from previous submissions at `http://www2.warwick.ac.uk/fac/sci/dcs/research/em/publications/`

web-em/. A quick scan of the models in WEB-EM-9 shows a deluge of macro use, ideal targets for LISP's unique power.

2 Provisional weighting.

Due to this study being a comparison of two different programming styles from the view of improving Empirical Modelling's ease of use, the models themselves will not be original, and hence I would like to put a provisional weighting of 70-30 towards the written part of the assessment. Should the modelling part of the weighting include marks from writing the interpreter itself, however, I would like to weight the marks at 40-60, with only 40