

Interpreter Internals: Unearthing Buried Treasure with CXXR

Andrew Runnalls¹

1. School of Computing, University of Kent, UK, A.R.Runnalls@kent.ac.uk

Keywords: *R*, CXXR, C++, packages, bignum

CXXR (www.cs.kent.ac.uk/projects/cxxr) is a project to reengineer the interpreter of the *R* language, currently written for the most part in *C*, into C++, whilst as far as possible retaining the full functionality of the standard *R* environment, including compatibility with existing *R* packages (cf. Runnalls, 2010). It is intended that by reorganising the code along object-oriented lines, by deploying the tighter code encapsulation that is possible in C++, and by improving the internal documentation, the project will open up new avenues of development for the *R* project. Development of CXXR started in May 2007, then shadowing R 2.5.1; at the time of this abstract it reflects the functionality of R 2.12.1.

From the beginning, different kinds of *R* object have been represented within the CXXR interpreter using a C++ class hierarchy, with the intention that this hierarchy could be extended by *R* developers. This reflects two strategic objectives:

- Given a C++ class (provided for example by a third-party library) it should be straightforward to introduce an *R* (CXXR) class wrapper around it, so that the functionality of that class can be exploited within the *R* language framework.
- Conversely, given an existing *R* class (provided typically by an *R* package), it should be straightforward to migrate its functionality—to whatever extent desired—into a new underlying C++ class.

Unfortunately, much C code within the standard *R* interpreter that could greatly support these objectives is not offered to packages as an API (Application Program Interface), and is not documented to API standards. Moreover, in large measure this code is hard-wired around the built-in *R* data types (REALSXP etc.), and so cannot easily be applied to other C++ data types.

This paper will explain how CXXR is endeavouring to change this situation by rewriting key pieces of interpreter code at a higher level of abstraction (often using C++ templates), and making it available to package writers via carefully documented interfaces.

Current progress will be illustrated by considering how to implement a package allowing the use of ‘BigInts’ (integers of arbitrarily large magnitude) within *R*. This is a task that has already been ably accomplished, building on the standard *R* interpreter, by Lucas *et al.* as part of their *gmp* package, which utilises GMP, the GNU Multiple Precision Arithmetic Library. But the paper will show how the CXXR framework makes it very much easier to establish key elements of the necessary functionality.

References

- Andrew Runnalls (2010). CXXR and Add-on Packages. <http://user2010.org/slides/Runnalls.pdf>. Presented at useR! 2010, Gaithersburg, MD.
- Antoine Lucas, Immanuel Scholz, Rainer Boehme and Sylvain Jasson (2010). Package ‘gmp’. <http://cran.r-project.org/web/packages/gmp/index.html>.
- Torbjörn Granlund *et al.* (2010). The GNU Multiple Precision Arithmetic Library. <http://gmplib.org/>