

Implementation of the BOLARE Programming Language

Viktor Pavlu
(vpavlu@complang.tuwien.ac.at)

*Institute of Computer Languages, Vienna University of Technology,
Argentinierstrasse 8/E185, 1040 Wien, Austria*

ABSTRACT

REBOL is a very flexible, dynamic, reflective programming language that clearly differs from the dynamic languages currently in popular use. The idea behind REBOL is that different problems should be attacked with different languages that have varying levels of granularity, each specifically tailored to its problem domain. These domain-specific languages should give the programmer the power to write programs that are closer to the problem and more expressive, thus shorter and easier to read, maintain and extend than would be possible in today's dynamic programming languages.

Despite its promising features, REBOL has not gained wide acceptance. Above all, this is due to the following: (1) The language is merely defined by means of its only implementation, (2) this implementation is closed-source and has many flaws, and (3) the flaws are here to stay.

We therefore started project BOLARE. Aim of this project is to build an interpreter for a language that very closely resembles REBOL but leaves out all unintentional behavior that only stems from artifacts of the original implementation.

After an introduction to REBOL/BOLARE in general, the three areas currently being worked on are presented:

Parser Generator: REBOL has a large set of built-in datatypes that have their own literal form making it easy to embed dates, times, email-addresses, tag structures, coordinates, binary data, etc. in scripts. The BOLARE parser is dynamically generated from a declarative description of literal forms. This allows the implementor of a new domain-specific language to extend the parser with custom literal forms during program execution.

Foreign Function Interface: REBOL has a rich standard library built into the interpreter but no means of accessing other libraries. BOLARE uses its foreign function interface (FFI) for C libraries to implement the essential parts of the standard library, but also to access arbitrary other C libraries.

Just-in-time Compiler: REBOL's flexibility comes at a price: the language cannot be statically analyzed and compiled. In BOLARE, too, there is no single syntax or semantics of the language, but values of different datatypes. Their meaning depends on the particular context in which the values are evaluated. We are currently trying to apply just-in-time compilation to expressions frequently evaluated under the same context.