OMeta Language

Who? Oleg Sivokon

When? *<2016-05-22 Sun>*

Outline

Dedication

Yes, but what is GlooP? If FlooP is BlooP unchained, then GlooP must be FlooP unchained. But how can you take the chains off twice? How do you make a language whose power transcends that of FlooP?
– *Douglas R. Hofstadter. Gdel, Esher, Bach: An Eternal Golden Braid*

Goals

- Short and concise description of a program.
- Minimalistic framework for writing compilers.
- Extending existing languages.

Implementation

- Unambiguous parsing (PEG vs LR/LL).
- Grammars as objects: inheritance, composition.
- Parametrized rules.
- Rules with free-style handlers (vs `visitor` pattern).

Compared to macros

- Hygiene vs scope management.
- Structured reader-macros.
- No need for symbol-macors and macrolets.

Compared to compiler compilers

- Full access to the underlying programming language.
- No need for lexers.
- No shift-reduce conflicts.
- Meta rules.

Compared to DCG (Prolog)

- Polymorphism due to OO features.
- Meta-rules in Prolog may be more expressive.

Credits and links

- Developed at Viewpoints Research Institute by Alessandro Warth and Ian Piumarta.
- Original implementation in SmallTalk.
- Implementations for experimental languages like Cola.

Industrial implementations

C#	ometasharp.codeplex.com/Wikipage.
C#	ironmeta.sourceforge.net/.
Python	launchpad.net/pymeta
Python	www.allbuttonspressed.com/projects/pymeta
Ruby	github.com/aquasync/ruby-ometa
JavaScript	github.com/alexwarth/ometa-js
Scheme	www.lshift.net/blog/2008/07/01/ometa-for-scheme
Common Lisp	subvert-the-dominant-paradigm.net/blog/?p=23
Clojure (not quite)	github.com/ericnormand/squarepeg