

Scripting Languages Dagstuhl

Talks

Semantics & Empirical Results

Gregor Richards: Eval begone!

- PhD student of Jan Vitek
- analyze use of eval in various scripting languages

Jan Vitek: The design of R

R : language for statistics

- Vectors, function, lazy and dynamic
 - looks a bit like a crazier APL
- Various object systems built on top of vectors
- Values always copied, so no aliases
 - but you can access variables from enclosing scope!
- promises are R's lazy values
 - evaluated as soon as you breathe on them
 - R is an eager, lazy language!

Analyzes performance etc

R is extremely slow and inefficient

Philippa Gardner: Reasoning about JavaScript

- Using operational semantics to reason about modifications to the store
- Separation logic

Ben Lerner: Language support for third-party code extensibility

Web browser extensions inject scripts into existing code in the browser

same privileges as browser code

JS mechanisms to support this

wrapping

redefine function that does new stuff and then calls original

monkey patching

grab source code

patch it

eval it

Problems

Both approaches break aliases

aliases are everywhere

Monkeypatching breaks closures

Need better support

dynamic aspect weaving

Stefan Hanenberg: Empirical studies on Static vs Dynamic Type Systems

experiments with students

Could not detect any significant difference in development time

Several experiments with inconclusive results

Arjun Guha: Engineering a JS semantics

LambdaJS models core JS as Flatt Felleisen calculus

desugars JS to LambdaJS

does $\text{desugar}(\text{JS-eval}(e)) = \text{LambdaJS-eval}(\text{desugar}(e))$?

influence by smalltalk, scheme ...

essentially: oo + events

block closures as event handlers

futures

Theo D'Hondt: AmbientTalk

Scripting languages: top 5

| | REPL | AT | Py |
|-------------------------|------|----|----|
| expressive, -pithy | ✓ | ✓ | ✓ |
| "boxed" process/service | ± | ± | ± |
| self contained | ✓ | ✓ | ✓ |
| paradigm/type agnostic | ± | ± | - |

scripting languages: top five

REPL

interactive, dynamic

implies multi-threading

expressive, not cryptic

like a DSL, close to domain, not implementation

boxed process/service

coordinate external black box components

related to auto-boxing: turn external things into manipulable components

self contained

can express everything in itself

can build debuggers etc

AmbientTalk is not self-contained since it depends on Java

paradigm/type agnostic

Should be able to use any paradigm

Python forces you to be OO

concinnity

Dave Herman: Life after main()

Static features matter too!

static scoping

macros, operator overloading

types

How to statically check stuff that refers to things that haven't been defined yet?

What's wrong with interfaces?

"Hope" was designed to deal with to checking stuff that hasn't yet been seen ...

Use modules to add new features over time ...

Meta-Programming & Security

Types and Contracts

Implementation, Modularity & Concurrency

Breakout sessions

Wild and crazy ideas

/ Scripting Languages Dagstuhl

<http://www.dagstuhl.de/en/program/calendar/semhp/?semnr=12011>

/ Scripting Languages Dagstuhl / Talks / Semantics & Empirical Results / Gregor Richards:

<http://www.cs.purdue.edu/homes/gkrichar/>

/ Scripting Languages Dags... / Talks / Semantics & Empirical Re... / Gregor Richards: / analyze use of eval in v...

<http://www.cs.purdue.edu/homes/gkrichar/papers/eval-ecoop-2011.pdf>

/ Scripting Languages Dagstuhl / Talks / Semantics & Empirical Results / Jan Vitek: The design of R

<http://www.cs.purdue.edu/homes/jv>

/ Scripting Languages Dagstuhl / Talks / Semantics & Empirical Results / Philippa Gardner: Reasoning about JavaScript

<http://www.doc.ic.ac.uk/~pg/>

/ Scripting Languages Dagstuhl / Talks / Semantics & Empirical Results / Ben Lerner: Language support f...

<http://www.cs.brown.edu/~blerner/>

/ Scripting Languages Dagstuhl / Talks / Semantics & Empirical Results / Stefan Hanenberg: Empirical st...

<http://www.dawis.wiwi.uni-due.de/team/stefan-hanenberg/>

/ Scripting Languages Dags... / Talks / Semantics & Empirical Re... / Stefan Hanenberg: Empiri... / experiments with student...

<http://www.cs.washington.edu/education/courses/cse590n/10au/hanenberg-oopsla2010.pdf>

/ Scripting Languages Dagstuhl / Talks / Semantics & Empirical Results / Arjun Guha: Engineering a JS semantics

<http://www.cs.brown.edu/~arjun/>

/ Scripting Languages Dags... / Talks / Semantics & Empirical Re... / Arjun Guha: Engineering ... / LambdaJS models core JS ...

<http://www.cs.brown.edu/research/plt/dl/jssem/v1/gsk-essence-javascript-r5.pdf>

/ Scripting Languages Dagstuhl / Talks / Semantics & Empirical Results / Theo D'Hondt: AmbientTalk

<http://soft.vub.ac.be/amop/>

/ Scripting Languages Dagstuhl / Talks / Semantics & Empirical Results / Dave Herman: Life after main()

<http://www.ccs.neu.edu/home/dherman/>

Scripting languages: top 5

| | AT | Py |
|-------------------------|----|----|
| REPL | ✓ | ✓ |
| expressive, ~pithy | ± | ± |
| “boxed” process/service | ✓ | ✓ |
| self contained | ± | ✓ |
| paradigm/type agnostic | ± | - |