Minutes of the: ***Ecma TC39-TG1***

held in: ***Phone conference***

on: ***31st March 2006***

Attendees

- Francis Cheng, Adobe Systems
- Jeff Dyer, Adobe Systems
- Brendan Eich, Mozilla Foundation
- Cormac Flanagan, UC Santa Cruz
- Gary Grossman, Adobe Systems
- Lars Hansen, Opera Software
- Dave Herman, Northeastern University
- Graydon Hoare, Mozilla Foundation
- Blake Kaplan, Mozilla Foundation
- Edwin Smith, Adobe Systems

Agenda

- ECMA rules and roles
- Dave's progress on spec language
- Lars' questions about scope

Notes

- ECMA Rules and Roles
  - pragmatic standards approach
  - invited expert definition

- Proposals marked with (!) need attention

- Lars' proposal on scopes
  - Lars' questions about scopes for directives and definitions
  - Package questions: yes (name visible only in def'n and lexically after), and yes (no nesting)
    - Why braced body if no nesting?
  - Classes and interfaces visible in entire prog, but no fwd refs in extends clauses
  - Namespaces: no hoisting to top of function, but hoist to top of block
  - Use directive: lexical (block) scope, pragma (compile-time directive)
  - Import is same as use, but what about switch cases or other conditional control flow?
    - Lars proposes restricting to top-level prog/pkg/class/func
  - Function: class code can access only statics
    - Sidebar on whether let x = y; should hoist x in enclosing block – resolved: yes
  - Revisit use on account of hoisting symmetry: should

```
let x = 42
{
```

```
    print(x)
  use namespace N
  print(x)
}  // where N::x = 33
```

print 42 and then 33, or 33 twice?

- Should import and use be allowed only first in a prog/pkg/class/func/block? Yes.
- namespace N is just like let N
- Motivation for namespaces from Flex SDK
- Dave: complexity from number of scope management tools gives pause
- Jeff will write namespace use-cases page in wiki
- Brendan to send osteele and micah writings on namespace usability probs

- Type system
  - Dave: working on formal notation for semantics, believes rigor will help.
  - Experimenting with Stratego (older page) and Maude.
  - Has fair amount of core Edition 3 in Stratego.
  - http://www.ccs.neu.edu/home/dherman/javascript/tg1/formal-semantics/
  - Stratego example: http://www.ccs.neu.edu/home/dherman/javascript/tg1/formal-semantics/stratego/ES-eval.str.html
  - Brendan notes Reference type history, advocates eliminating from ES4 in favor of LHS grammar specialization
  - Maude may allow provers to be hooked up
  - Lars: Haskell instead, for a type system? Dave: have to invent pattern-matching, hook up model and meta languages
  - Cormac: accessible to audiences of ES4 spec? yes, we think – aspire to make it so
  - Cormac: dependency chain from runtime semantics to type system to contract system

- Spec fixing
  - Lars and everyone to update bugs