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

held in: ***Phone conference***

on: ***11th September 2007***

Attendees

First attempt:

- Francis Cheng, Adobe Systems
- Graydon Hoare, Mozilla Foundation
- Lars Hansen, Adobe Systems
- Dan Smith, Adobe Systems

Second attempt:

- Jeff Dyer, Adobe Systems
- Graydon Hoare, Mozilla Foundation
- Lars Hansen, Adobe Systems
- Pratap Lakshman, Microsoft
- Brendan Eich, Mozilla Foundation

Agenda

- Open proposals
  - Self type
  - Resurrected eval
  - Program units, note Graydon's edits / clarifications since last week
- Open tickets
- Schedule
  - Date for October/November meeting? (Thanksgiving in the US is on Nov 22nd.)

Minutes

Resurrected eval:

- Generally approved though some details are outstanding
- The spec takes over as the canonical description
- Lars to clean up the proposal page lightly to note outstanding issues

Self type:

- Brendan: I think "self" probably ought to mean the directly enclosing nominal type
- Cormac: Then you could just use the nominal type itself
- Brendan: But using the allocated type has some utility problems
- Cormac: Yes, but it's for the sake of subtyping

- Brendan: Maybe self doesn't belong in the nominal type system. Does `self` in a nominal type setting mean exact type or "that type or subtype"?
- Cormac: Usually subtypes are included...
- Brendan: how about not allowing it in nominal types – it would be a hazard
- Lars: makes sense to me, the nominal type system doesn't need it
- Cormac: Self types are a bit researchy in either setting...
- Brendan: But Bruce's work goes back to 1995

- What is the utility of 'Self' in a nominal setting? The use case in the "map" proposal is not obviously sound, it doesn't provide a useful bound like it was intended to
- Lars will tidy up that bit.

- Cormac: so which are the interesting / useful structural types that might use "self"?
- Brendan: iterator/generator, clone – `this:self`, function returning `self`. Argument `self` is more iffy.

Program units:

- Jeff wonders what the motivation is for not reporting verify errors if a type is missing
- Graydon: that's the point of the exercise, verify errors can't be reported at that point, there can be more fragments coming in the future. If it's inside a unit the verifier can fail when the unit ends, it's only outside units this can't happen
- Graydon: there are some subtle issues. In standard mode one might even wish to report errors (like missing base classes) that will always result in errors at run time.
- Accepted, we'll commit to it and work out the small details later

Schedule:

- Mozilla hosts in September (27/28)
- Nov 8/9 2007, tentatively, host TBD
- Jan 24/25 2008, tentatively, host TBD