Minutes for the: *10th meeting of Ecma TC39*
to be held in: *Mountain View, CA, USA*
on: *28 - 29 May 2009*

1 Opening, welcome and roll call

1.1 Opening of the meeting (Mr. Neumann)

The meeting was opened by Mr. Neumann at approximately 10:15 AM on 28th May.

1.2 Introduction of attendees

John Neumann – Ecma

Douglas Crockford – Yahoo'

Kwans Yul Seo – Company 100

Erik Arvidsson – Google

Mark Miller – Google

Allen Wirfs-Brock – Microsoft

Sam Ruby – IBM

Waldemar Horwat – Google

Cormac Flanagan – UCSC

Dave Herman – Northeastern University

Brendan Eich – Mozilla

Rob Sayre – Mozilla

Chris Pine – Opera (Phone)

Istvan Sebestyen – Ecma (Phone part time)

1.3 Host facilities, local logistics

Rob Sayre described facilities, lunch, and dinner plans for the meeting.

2 Adoption of the agenda (2009/028)

Agenda was adopted as presented.

3 Approval of minutes from March 2009 (2009/022)

Minutes approved as presented. Minutes for SES (2009/023) approved as modified. Revised document will be sent to Ecma for re-publication.

4 Review of project candidate draft testing, including outstanding issues

IE ES5 Prototype is available to any TC39 member if requested from Mr. Wirfs-Brock. The prototype is a version of the ES 5 implementation that runs in IE 8. All conformance tests are available today from Mr. Wirfs-Brock and licensed under dual BSD and MS-PL open source licenses. It was previously agreed to do this at Mozilla. In general there are problems with

licensing and copyright issues for conformance tests. The MS test suite is simpler than Mozilla and easily plugged into the Mozilla Suite. There was agreement that the existing Mozilla Suite and the ES 5 test suite should be kept distinct for development purposes. The plan following testing is to formalize this activity inside a Technical Report. Parts of test prototype are not yet satisfactory to Mr. Wirfs-Brock (for example Strict Mode needs to be deeper, including Array Extra's).

The zip file of the conformance suite has been posted to Ecma site along with licenses. Hosting of the conformance suite development at Codeplex is under consideration but may require a joint agreement. Mr. Ruby points out that hosting at IBM in the past has been perceived negatively and as a result it did not work. Proposed dual licensing is a problem as members other than Microsoft do not necessarily want to use MS-PL license. It was agreed that all contributions must include a BSD license and that dual licensing using other licenses would be optional for each contributor. Mr. Wirfs-Brock will work on resolving the CodePlex copyright assignment and development agreement issues.

Mozilla implementation is progressing and should be available after Firefox 3.5 is released.

Mr. Pine described the current status of Opera Testing. Since he is on paternity leave he is not in touch with the activity but gave us a brief status of making progress.

A discussion of the outstanding issues in the errata document prepared by Mr. Wirfs-Brock was conducted and all open issues were resolved. Contribution from Mr. Horwat was also discussed and issues resolved.

Notes from Mr. Horwat:

The last SES meeting was the final one.

Microsoft has an IE8 plug-in that implements essentially all of ES5. This will be available to TC39 and hopefully the world at large as well.

Microsoft has a test suite under the BSD license but contributors to Microsoft's test suite repository must assign copyright to Microsoft. This means that Mozilla could grab snapshots but not contribute back.

Maybe put Ecma brand on the test suite instead ? Mr. Sebestyen would be happy to host ECMAScript test suites donated to Ecma.

Mozilla will have ES5 essentially implemented 4-6 weeks after the release of Firefox 3.5.

No word from Apple.

There's a student working on ES5 for Rhino.

Item 5 of the agenda refers to the discussion of Decimal at the Coordinating Committee meeting but there is no mention of it in the minutes of that meeting.

Mr. Sebestyen presented the happenings at the CC.

Mr. Miller: Given that Sun is getting acquired by Oracle, this may give us an opportunity to license the trademark for Ecma and rename ECMAScript back to JavaScript.

Google will host the December 2009 GA in Mountain View.

"use strict" resolution: A use strict directive must be the literal string "use strict". It must occur before any statement other than other literal string-semicolon pairs.

Unicode issues: Strings are raw 16-bit code units, source text is Unicode. Trouble occurs when a supplemental character occurs in a string literal -- the spec is not well-defined. Agreed to change wording of chapter 6 to state that source text is first converted to 16-bit code units if it's not already in that form. If it's in one of the Unicode formats then convert it to UTF-16 and treat the resulting 16-bit code units as inputs to the lexer grammar. If it's already in 16-bit code units (as would happen when eval'ing a string), pass those through verbatim. Note that they might contain unmatched surrogates which would not have been allowed under UTF-16.

If you want to treat a supplemental character as a regular expression atom, parenthesize it. You'd need to do it anyway for things like a-umlaut that contain combining characters.

Mr. Crockford: class const enum export extends import super are the only FutureReservedWords that are actually reserved on IE (or any other browser).

We'll unreserve the rest except for the following, which would behave just like let and yield: implements interface package private protected public static

8.6.2 paragraph before Table 5: Agreed to nail this down: Remove the first sentence and spell out the rules one by one. Note that the rules about [[Writable]]==false implying no value changes is false (the property can be redefined). The rule about [[Configurable]]==false implying no attribute changes is also false (the [[Writable]] attribute can be legally cleared). A rule about no new properties appearing on non-[[Extensible]] host objects is missing. A rule about not flipping [[Extensible]] from false to true is missing.

Agreed to delete both EvalError cases from 8.7.2. They're unnecessary and cause trouble.

Agreed to treat "arguments" definitions the same as "eval" definitions within strict mode. Both would cause syntax errors. There is no longer anything that throws EvalError but we will retain it for compatibility with ES3.

We'll extend the function definition grammar to make the name optional. If the name is missing, the function is not declared but instead it is returned as the completion value. This is needed in practice by the current browsers' implementations of eval.

Object.create will allow a null prototype. This is very useful for making pure hash table objects.

Failed writes in array functions will throw, as per discussion on the list.

15.4.4.11:
- Allow sorting non-extensible arrays if they have no holes
- Guarantee not calling delete when sorting an array that has no holes
- "this does not have a property with name ToString(i)": should be "own property"

Got rid of arguments.caller poison pill since browsers don't actually implement it.

Rather than describing the evil things that implementations do with F.caller, we agreed to just impose a blanket prohibition of code peeking into the environment records or identity of strict functions on the stack. This way a test suite can ensure that F.caller does not reveal strict functions without us having to introduce the evil things into the standard. Mr. Horwat will write up proposed wording.

5 Discuss and agree to forward going procedures for inclusion of Decimal in the next edition of ECMAScript, including results of CC meeting in May

The committee can agree on developing better library support in the language, then accommodate decimal through the library approach. Mr. Ruby has serious concerns about this approach since it has been discussed for a long time and no solution has been agreed and library components are not easily integrated into applications since the library might not be downloaded and made visible to the application and concern for efficiency exists as well. Mark Miller will begin investigation of library extensions but at this time there is no agreement on how or when decimal can be included in ECMAScript.

Detailed notes from Waledmar Horwat:

Decimal: There were significant technical issues with the proposal last Fall. Sam Ruby: nothing has changed since August. Mr. Wirfs-Brock: we haven't committed to anything for Harmony. The design rules for Harmony are different from those of ES3.1; Harmony takes a broader perspective of adding types.

Past decisions, as revised since last summer:
typeof 1.1m == "decimal"
1.1m === 1.10m
(1.10m).toString == "1.1"
(1e+3m).toString == "1000"
Mr. Crockford, Mr. Wirfs-Brock: All things are on the table; nothing is ruled in at this point. Decimal depends on having a type system.

Mr. Miller, Mr. Arvidsson: Would prefer not to have decimal at all. Failing that, we should solve the general operator extensibility problem first to make sure that decimal fits into that extensibility framework.

Mr. Eich: Decimal is a frequently-reported bug. Interaction between decimal and binary floats would be a source of constant headaches. The big red switch approach (of changing everything to decimal) would be a performance non-starter in this era of high-speed ECMAScript implementations. Also, the usability problem needs critical thinking; this is what killed decimal in ES4.

Mr. Crockford: The goal is not adding decimal to the language. The goal is to fix the usability bug.

Mr. Horwat: No big red global switches to switch modes. This makes code interoperability impossible. Note that "use strict" is not a big red switch: it's lexical and strict code readily interacts with non-strict code. With decimal modes interactions across the boundary would be a problem. The alternative would be to move the entire language over to decimal but decimal performance is woefully inadequate to consider it in the foreseeable future.

Mr. Wirfs-Brock: Others at Microsoft want other numeric types too. Decimal would open the floodgates.

Mr. Ruby: Goal is not solving the decimal usability bugs; the goal is simply adding decimal to the language.

Mr. Miller: Goal is to get committee out of the business of numerics; instead, provide facilities for libraries to implement those. Mr. Miller sent a proposal a few months ago.

Mr. Miller's proposal was (unknowingly) very similar to earlier ES4 and ExtendScript proposals, which weren't received well back then.

Mr. Wirfs-Brock: Consider numeric literals as their own types, which was discussed at a past meeting. A few groans followed.

Back-and-forth over whether decimal should be an (optional or mandatory) library.

Mr. Miller: IEEE 754r decimal is unattractive. We fixed the cohort problem. Not clear if it will be picked up by the processor manufacturers with any decent performance.

Mr. Eich: The committee cannot agree to require decimal as a normative feature of upcoming ECMAScript standards. The library approach might be an alternative.

Mr. Eich, Mr. Wirfs-Brock: Extension mechanisms are the more essential mechanisms. We should implement decimal using those first rather than picking a library winner today. The decimal library would be a good use case.

Mr. Ruby: Users don't generally download binary extensions (with the exception of flash). An ECMAScript-only library would be too slow.

Mr. Miller: Library should produce type of (decimal) be Object. Mr. Horwat: would need to overload === if you want to take that route. This likely leads to the concept of value types. Mr. Miller will make a strawman proposal of value types with double dispatch operator overloading and put it on the wiki.

The committee has not agreed on an approach yet. Things are in an exploratory phrase, with a lot of dragons to beware of.

6 Discussion of ES harmony and SES and their relationship

WIKI page lists rules for SES design experiments:

Looking at using ES 5 as a base and how that relates to ES Harmony.

Harmony discussion:

New Specification and Reference Implementation (last harrah for old spec)

- Lambdas PH
- Let, const, function *
- Lexical_scope *
- Names
- Module System *
    - Contracts PH (maybe)
    - Macros PH
    - Types PH (maybe)
- Operators and literals -> value types
- Weak references
- Catchalls
- Iterators
- Generators
- Rest *
- Spread *
- Default parameter values *
- Hermetic eval, spawn *
- Classes
- Return to label PH

* are firm candidates for first Harmony version

? status unknown for first version

PH is known post harmony version 1

Notes from Mr. Horwat:

Microsoft's conformance suite will be dual-licensed under BSD and MSPL (similar to Apache).

Curiosity about how Ecma licenses software (as opposed to documents).

Harmony list:
New spec with a reference implementation
Lambda (plenty of opposition)
Let, const
Lexical scope
Names
Modules
Contracts
Macros

Types
Catchalls
Iterators
Generators
Rest
Spread
Default parameter values
Classes
Return to label
Weak references
Extensible operators
Extensible literals
Value types
Hermetic eval

Significant issues with catchalls due to climbing the meta ladder.

Macros are most likely post-Harmony.

Discussion of static local variables. Should these create global variables or inject variables one lexical function level up?
Global variables are the usual way of doing it but Mark objects that they break the modules proposal by injecting a covert communication channel. Mr. Horwat objects to just hoisting static variables one function level up because it breaks simple refactorings:

```
function foo() {
  static var ...
}

function factory() {
  ..
  return foo;
}
```

refactored to:

```
function factory() {
  function foo() {
    static var ...
  }
  ..
  return foo;
}
```

would be broken.

Mr. Horwat: Possible solution ideas: Label the scopes and allow variable declarations to inject variables into lexically outer labeled scopes but with the variables only being visible in the scope of the definition (unlike, say, var hoisting, which makes the variable visible outside the block in which it's defined).

Mr. Miller's name and ephemeron equivalence:

```
const n = new Name(...);
x[n] = y;
x[n];
```

is equivalent to:

```
const n = new EphemeronTable(...);
n.put(x, y);
n.get(x);
```

We'd want a Name n to be garbage-collected if the only use of n is in hidden places such as x[n]. That would also collect y if it's not reachable from anywhere else.

Mr. Miller: Names and let allow us to have both class-private and instance-private.

```
class Point {...}
pt: Point = new Point(...);
pt = somethingElse;
```

The "pt: Point" annotation puts a guard on the slot pt that ensures upon every assignment to pt that Point.accepts(pt). accepts should be monotonic (i.e. if const x = ... and Point.accepts(x) is true now, then Point.accepts(x) must be true later).

```
const [brander, guard] = Brand();
const pt = Object.ccreate(..., ..., [brander])  // assigns zero or more brands
var pt2: guard = pt;  // passes test
```

Lots of agreement on classes: zero inheritance, nominal typing, tied in with type guards, easy and usable syntax, a class definition is a run-time construct that creates a new class each time it's executed (no compile-vs-runtime phase separation).

Agreed to make function substatements (not at the top-level block of global or function scope) are syntax errors in ES5 strict mode.

7 Next steps

1. For ES 5, update spec reflecting this meeting and complete testing and final set of technical changes to be agreed at next meeting
2. For ES Harmony concrete proposals for specification of included functions from above list
3. For Decimal, Mark Miller will generate proposal of value types with double dispatch operator overloading and put it on the wiki.

8 Date and place of the next meeting(s)

- July [29-30] (Redmond, Washington) @ 10 AM [Microsoft]
- September [17-18] (Boston w/AJAX) @ 10 AM

  TC39 will meet at same hotel as AJAX and Mr. Eich will give me hotel information so Ecma can make reservation and meeting support.
- November [5-6] (Santa Clara, California) @ 10 AM [Ecma – co located with W3C]

  TC39 will meet on Thurs/Fri at same hotel (Santa Clara Marriott). Groups we want to avoid overlap with include HTML5 and DOM. We will send general invitation to W3C members to attend TC39 meeting. TC39 members who want to attend W3C must be company members of that group and be identified by their W3C member as delegates to the meeting. Otherwise attendance will not be possible.

9 Closure

The meeting closed at approximately 4:30 on 29th May.