Minutes for the: *16th meeting of Ecma TC39*

held in: *Mountain View, CA, USA*

on: *24- 25 May 2010*

1 Opening, welcome and roll call

1.1 Opening of the meeting (Mr. Neumann)

The meeting (hosted by Google at their premises in Mountain View, CA) was opened by **Mr. Neumann,** Chair of TC39 at approximately 10:15 AM on 24th May 2010 (2010/022: Venue for the 16th meeting of TC39, Mountain View, May 2010).

1.2 Introduction of attendees

The meeting participants introduced themselves.

2010/023 Project proposal regarding test suite for ECMA-262 5th edition by Mr. Wirfs-Brock

2010/025 Technical Report on interoperability/conformance tests (to be posted asap)

1.2.1 Ecma participants present at the TC39 meeting

John Neumann – Ecma – TC39 Chair

John Mitchell – Stanford University

Oliver Hunt – Apple

Brendan Eich – Mozilla

Sam Tobin-Hochstadt – Northeastern University

Andreas Gal – Mozilla

Waldemar Horwat – Google

Cormac Flanagan – UCSC

Narj Nuikker – Google

Allen Wirfs-Brock – Microsoft

Ken Russell – Google

Vladimir Vukiceul – Mozilla

Jeff Dyer – Adobe

Arun Rangarathan – Mozilla

Erik Arrvidsan – Google

Ihab Awad – Google

Douglas Crockford – Yahoo!

Sam Ruby – IBM

Gavin Barrraclough – Apple

Istvan Sebestyen – Ecma-International – TC39 Secretary (phone, partly)

Tom van Cutsem – Google (phone, partly)

1.3 Host facilities, local logistics

Mr. Miller welcomed the delegates on behalf of the host Google and provided some useful information on the local logistics and on the Social event.

2 Adoption of the agenda (2010/024)

The agenda Ecma/TC39/2010/024 Agenda for the 16th meeting of TC39, Mountain View, May 2010 (Rev. 1) was accepted with minor additions (add: 4.7 and 4,8):

- on the status of "decimal" will briefly reported,

- future plans on ECMA-357 (E4X) will be discussed.

- Agreed to one hour joint meeting with Khronos for 3PM on Tuesday.

3 Approval of minutes from March 2010 (2010/021)

The minutes of the March 2010 TC39 meeting have been approved unanimously without any changes. Noted that they lacked technical details of some technical discussions. Minutes for this meeting augmented by Waldemar Horwat for technical content.

4 Report from the Secretariat

4.1 Report from the CC meeting, 28-29 April 2010

Mr. Sebestyen has reported about the recent Ecma CC meeting that took place at the Ecma Office in Geneva. There, the TC39 chairman's report 2010/020 was presented by him. The CC was pleased with the progress of work of TC39.

Mr. Sebestyen also said that he reminded some CC representatives that if their company really wanted to influence the ECMAScript standardization then it is not enough to observe the TC39 standardization activities from remote, but they have to participate in the TC39 work actively.

Mr. Sebestyen also reported that regarding the statistics on download of Ecma standards over the Internet there is no change, ECMA-262 is still the most popular standard in Ecma, and download figures are basically the same as last year. The number of downloads still account for 30-40% of all Ecma downloads.

Mr. Sebestyen also brought up the point that a solution to **"External contributions to Ecma Work"** was discussed by the CC and approved.

This is required by some Ecma TCs, where they expect significant external contributions from non-Ecma members. There the issue of IPRs is important. Simple questions by external people, obvious comments, or pointing to typos in an Ecma standard does not belong into that category. Those would be dealt with by the usual feedback mechanism (e.g. email) to Ecma.

This external contribution practice will provide a right form for external contributions from non-members. Those are not bound by the Ecma by-laws, so at least the same policy is needed. If such a feed-back is coming there is always a need for registration via the Ecma Website. This is always linked to a specific Ecma TC and project.

Applicant contributors sign up for a specific project, using this agreement text:

> *"By requesting registration to <Ecma project>:*
>
> - *I confirm that I have the right to provide all contributions I will make, and if I am making a contribution on behalf of an organization, I confirm that I have the right to make this contribution on behalf of my organization and to bind my organization to the obligations below*
> - *I (or my represented organization) grant Ecma the irrevocable right to use, free of charge, all or parts of my contributions for the purposes of standardisation and, in case the implementation or use of the standards incorporating the feedback contribution requires*

> *patents or any other kind intellectual property rights that I (or my represented organization) own, I (or my represented organization) will grant licenses to other parties on non-discriminatory basis on reasonable terms and conditions.(here link to the Ecma Code of Conduct on Patent Matters)"*

The CC has approved this, and instructed the Secretariat to implement this. Currently TC38, TC45 and TC46 are using this tool. TC39 should simply take it into account, if it needs such kind of tool in its project or not.

5 General technical and legal matters

5.1 Discussion of Licensing/Copyright/Trademark matters

On the **Trademark** of ECMAScript there were no new developments reported.

On **Software Copyrights Mr. Sebestyen** has reported that the CC Ad-hoc group (interested CC members and invited legal experts) on Software Copyrights has completed its work and has submitted the proposed software copyright policy to the Ecma CC. As has been stated in the past the policy is an experimental policy, and it is to be used by TC39. Based on the TC39 experiences it will be decided if also other TCs will use it in the future. That input document version to the CC meeting was also distributed for information to TC39 earlier.

The Ecma CC has discussed the document, and made minor editorial changes. The changed document has been published as GA and TC39 documents (*2010/026 Ecma International Policy on Submission, Inclusion and Licensing of Software).* The document is up to GA voting at the Ecma GA on June 17, 2010. It is expected that the policy will be approved by the GA, since there was full consensus by the Ad-Hoc Group and by the CC. Therefore TC39 is quite safe to apply it with immediate effect.

Then the software copyright policy was briefly explained to TC39 by **Mr. Sebestyen** and **Mr. Wirfs-Brock**. It is basically a policy where the final copyright in Ecma Standards and TR belong to Ecma International (as usual). Ecma International grants free license for implementation, use, further development according to the so called "New BSD" license. Software submitters have their own free choice to submit their software for Ecma standardization either under the BSD license or a more restricted license for the purpose of standardization only (thus until approval of the standard).

TC39 has taken with satisfaction note of the report. **Mr. Neumann** has asked **Mr. Sebestyen** to extend to the CC the gratitude of TC39 for the speedy delivery of the Software Copyright Policy. He asked to convey this message both to the CC and to the GA.

5.2 ES5 JTC 1 fast-track status and ES5 Erratum

Mr. Wirfs-Brock has reported that the current best Erratum is on the ECMAScript Wiki (http://wiki.ecmascript.org/doku.php), with a status of May 10, 2010 (The document has been posted as 2010/027). That document is not using the JTC1 required comment template yet. TC39 members have been invited to look at it and to submit further comments. He promised that by the next meeting of TC39 this will be turned into a document according to the JTC1 comment template. That document needs to be approved by TC31 and submitted to JTC1 (SC22) by the Ecma Secretariat. The final document to be approved by TC39 will be published and distributed in time for the TC39 July meeting.

Mr. Sebestyen reminded the meeting that the 5 months technical comment period will end on August 6, 2010. Thus it is very important that the Ecma TC39 comments get there by this date. Depending on the comments received there might be a BRM for the standard in form of a BRM meeting, or if the comments are trivial then just the International Editor (**Mr. Wirfs-Brock**) will make the changes and publication of the standard will go ahead.

ES5 erratum highlights:

- String indices subject to 2^32 limits and they shouldn't be?

- The committee agrees that nonenumerable properties should shadow enumerable ones when enumerating. Some think that the standard text already states that, others think it could use slight clarification.

5.3 W3C discussion – Status of potential joint meeting with W3C in November 2010

There was a short discussion on the further collaboration with W3C especially a possible TC39 joint meeting with W3C in November 2010.

Mr. Neumann said that basically he has nothing new to say. He contacted W3C (**Philippe** Le Hegaret), but no reply from W3C. This indicates no interest. **Mr. Wirfs-Brock** will try to investigate the situation the next week. We will come back this issue later.

Mr. Neumann asked that if W3C is interested in “Modules”, will TC39 be in a shape in November 2010 to talk about “Modules” with them.

Primary area of concern: WebIDL and its ECMAScript language binding.

Summary, as we can judge today, we will not meet with them.

5.4 ECMA-357 (E4X) update or revision

It was pointed out that the current version of the standard is being shipped by some current products, so the standard cannot be withdrawn right now – even if it only relates to ES3.

ECMA-357 should be revised or obsoleted by December 2013. It has some problematic features such as its with-on-steroids. Maybe ECMA-357 can be "proxified" to be implemented with Harmony. Some of its features would need to go to make that happen, such as its distinction between method get and method call.

There was a discussion again on the faith of ECMA-357 in the May 2010 meeting. It was then agreed by TC39 that ECMA-357 either needs a significant update or withdrawal. ECMA-357 will, however, not be updated to bring it into harmony with ES5. It was then agreed that ECMA-357 will be updated later and in close harmony with the next Edition of ECMAScript in December 2013. So, it would be compatible with “ES-Harmony”.

5.5 Project Proposal on the development of an Implementation Test Suite for ECMA-262, fifth Edition.

Allen Wirfs-Brock has summarized the project (2010/023), which is the continuation of the current projects, but slightly modified in light of the new developments that TC39 can apply new policies (such as Software Copyright):

> *Project Summary: TC39 will undertake the development of a test suite that verifies the correctness of an ECMAScript implementation relative to the ECMA-262-5 (ES5) specification. The intent is for the completed suite to fully test all aspects of ES5. In addition to the actual tests, this project will include development of a web-based test page that will run the tests for browser-based ECMAScript implementations. When completed, the test suite will be “published” as an ECMA International Technical Report and TC39 will recommend that the final web-based test page should be hosted and accessible from the Ecma International web site. In addition, this test suite and its supporting infrastructure shall serve as the basis for testing and validating features under consideration for future editions of the ECMAScript specification.*
>
> *Background: The topic of conformance testing has been under active discussion within TC39 for several years and TC39 even set a goal (that was not achieved) that the fifth edition should pass conformance testing by multiple implementations prior to its final publication. Until recently there were simply no comprehensive test suites available for ECMAScript. Within the last year two open-source efforts relating to such testing have appeared: Sputnik, a comprehensive test suite for ES3, was developed and released with the support of TC39 member Google; ES5Conform, a suite that focuses on testing new features of ES5, was developed and released primarily with support of TC39 member Microsoft. Both of these suites have become valuable resources to the*

ECMAScript development community. However, there currently is no unified test suite that covers all of ES5. Also, while there is no evidence that this is actually the case, occasionally concerns have been expressed that there may be something partisan about test suites that are primarily developed and hosted by single organizations. When this issue was last informally discussed within TC39 the primary blocker was the lack of suitable Ecma International IPR policy that could be applied to this type of software-based project. The Ecma Coordinating Committee has now approved such a policy and it is anticipate that final approval will be granted by the Ecma General Assembly at its June 2010 meeting. This proposal is made in anticipation of that approval.

Project Goals: Create a comprehensive test suite for can be used to determine whether an ECMAScript implementation correctly implements all aspects of the ES5 specification. The suite should consists of individual tests and test driver that can be adapted by implementers to their specific development situations and also a web-based test driver that can be used by the general public to test browser-based ECMAScript implementations. The test suite will be published as a software-based technical report under the Ecma Software licensing policy (a BSD license).

Project Organization: The development of the test suite will be a subproject within TC39 but will not be organized as a distinct Task Group. All TC39 members are invited to participate in the project. In addition, TC39 will designate one or more individual TC39 participants to serve as "project editor". Since this project is a software development activity, the project editor(s) will serve as the "lead developer" of the project coordinating the design and implementation of the technical content of the project. The project editor(s) will report back to TC39 at its regular face to face meetings the status of the development effort.

Project IPR Policies: The project will operate under the new TC39 software licensing policy. This requires all participants and the organizations to execute a contributor's agreement that donates the software to Ecma and licenses it using a New BSD license. All contributions must come from or through TC39 members. The software licensing policy gives contributors the option of making their contributions immediately available under the BSD license or deferring BSD licensing until final GA approval and publication. Ideally, this project will operate with full public transparency during its development. For this reason, members are encouraged to use the immediate BSD licensing alternative for their contributions.

Software Contributions: TC39 members are invited to donate their existing conformance test suites and supporting infrastructure to the project. In addition, members are requested to actively contribute to the ongoing development effort for this project.

Infrastructure: The development project will be hosted on the ecmascript.org server using the bug tracking system (http://bugs.ecmascript.org) and the Mercurial Repository (http://hg.ecmascrpt.org) that are already operational. During development, the test runner web page will also be hosted and publically accessible at ecmascript.org.

Project Schedule: Final schedule will depend upon what member contributions are initially developed and what new materials are determinate by the project editor(s) to be necessary to supplement those contributions. Ideally, the initial contributions should be made publically available via the ecmascript.org site as soon as possible. It is TC39's desire that this project be completed in time for GA approval in June 2011 (April, 2011 completion) at the latest.

This project had been previously planned by the TC39 but was on hold pending completion on an adequate Ecma software licensing policy. Based upon the CC's policy approval, TC39 agreed to proceed with this project as described in document 2010/023. **Allen Wirfs-Brock** will be interim project editor.

The conformance test suite will likely be an ongoing project rather than stopping at the TR.

It was agreed by the meeting to continue the project under these terms. Microsoft and Google will provide tools in the next months to the project. Call to other TC39 members has been made.

The goal would be to have the first TR ready for GA approval in June 2011. That means that Ecma TC39 has to finish it by End of March – Early April 2010. We need commitment on contributions from TC39 members by the next meeting of TC39. MS is thinking about moving its components to the Ecma Website and making the necessary conversions.

6 Discussion of ES harmony

6.1 Progression of ES-Harmony

6.1.1 Specification Language

Debate about whether using ECMAScript for the specification language is appropriate. Some issues:

- How to define "if" in terms of "if", "<=" in terms of "<=", etc. Much of the language would be circular in this way.
- Many opportunities for accidents. The "+" operator can add numbers, concatenate strings, etc., and it's not clear which one is being used in pseudocode. One might accidentally concatenate instead of adding when specifying an algorithm.
- Confusion between the specification language and the specified language. Counterpoint: Use ES5 to specify the specification language. Counter-counterpoint: Shouldn't refer to obsolete versions of the standard normatively. Also want to be able to express new data structures from within pseudocode.

Allen Wirfs-Brock and **Waldemar Horwat**: Useful to have a reference implementation in some specification language and develop that in parallel with the specification. However, the specification need not be written in the same language. One way to do this is to have a single source S and mappings from S to an executable reference implementation (not part of the spec) and from S to the spec. The latter mapping is not objective.

Waldemar Horwat will revive the Common Lisp semantic code, particularly the parser and grammar checkers by the next meeting.

Consensus: The status quo seems attractive to many in the committee. **Oliver Hunt**: the status quo pseudocode in ES5 worked better for him as an implementor than any code he's seen.

6.1.2 ES 5 (Harmony) Definitional Interpreter

No discussion. Resolved under discussion of specification language above.

6.1.3 ES 5 to ES Harmony Transition Strategies

Allen Wirfs-Brock: Would be helpful to clean up chapter 8.

6.2 Discussion of ES harmony themes/appointment of managers

6.2.1 Division of work

Modularity, Isolation, Virtualization, Control effects, Library/tool enabling, Language reform, and Versioning

Waldemar Horwat: Key requirements for classes: Unforgeable trademarking, object shaping, *not* creating O(n*m) objects for n instances with m methods.

Great debate over whether modules or classes should be the top priority for Harmony

6.2.2 Concept of Theme managers/advocates and Proposal champions

Themes: Consensus with the idea of having themes and filtering proposals based on them, but disagreement on the specific choice of themes. Missing: abstraction/data structuring/classes, internationalization.

Submit wiki write permission requests to **Brendan Eich**.

Discussion about classes. **Mark Miller** will revive the proposal.

7 Additional Discussions

7.1 Ephemeron Tables

Naming discussion on EphemeronTable: It shouldn't be called WeakKeyTable because the latter has a well-defined meaning--it can't collect cycles like {Key1->Key2, Key2->Key1}.

Mark Miller's ephemeron tables conflate storing undefined as a value with not having a binding at all. One deletes a binding by setting its value to undefined. This means that the example of walking the prototype chain doesn't work for undefined values.

Ephemerons retain the invisibility of garbage collection. Enumeration of ephemeron tables (currently disallowed) would make gc observable.

Allen Wirfs-Brock: Levels of impact on gc (and therefore on system performance):
1. Identity-based hash tables: none
2. Weak-keyed tables: more global impact; special processing
3. Ephemeron tables: larger impact; addresses cycle problem

Waldemar Horwat: There's also a level 0 (class-private fields), which would have even less impact because it does not need an identity-based hash table at all. Level 0 is sufficient to implement things like the money/purse ephemeron example.

Question: Do the extra costs of ephemeron tables over weak key tables appear even if there are no cycles that an ephemeron table would collect but a weak key table wouldn't? Answer: No; the two tables should have equivalent performance in that case. However, it is possible to carry out denial-of-service attacks by creating pathological ephemeron cycles.

Ephemeron tables dominate weak-key tables. There's little reason to have the latter if we have the former. Identity-based hash tables might be useful on their own.

Agreed to move the status of ephemeron tables from strawman to proposal. Implementors reluctant with trying strawmen may now experiment with implementations of ephemeron tables to try to figure out what issues would arise. The two open issues are delete-vs-undefined and the name of the abstraction.

7.2 Dr. SES (Dr. SES = Distributed Resilient Secure ECMAScript)

Waldemar Horwat will revive the Common Lisp semantic code, particularly the parser and grammar checkers by the next meeting.

Mark Miller: No need for new in-band opt-ins. Might want an out-of-band opt-in to keep old browsers from trying to execute "let" statements and such. **Mark Miller** is concerned about existing browser practice with named nested function declarations. There is no common semantics. **Brendan Eich**: Mozilla will change to conform to whatever the committee reaches consensus on in such cases.

Mark Miller will do a write-up for const and let variable declarations and nested named functions. We've had a verbal agreement on these for a long time.

Past agreements: Functions are hoisted to the top of a block and initialized before the block begins execution. Functions are let-bound, not const-bound. Consts are block-scoped and have a temporal dead zone where accessing them would cause an exception.

7.3 A possible generalization of Proxies

Mark Miller: Proposed proxy changes:

1. x[y] would call the get method that would not auto-coerce y to a string. Questions about impact of this on performance: when do implementations intern strings for property access? Implementors seem to all do the coercion/interning relatively early, so thing change would be inconvenient. Could this have an effect on the order in which multiple user methods (such as coercions) can be run as side effects of evaluating an expression? **Mark Miller** withdrew this proposal after discussion.

2. Recursive fixing. What happens when fixing a proxy makes a recursive call to fix the same proxy? Proposed answer: Throw a type error on the inner call.

3. Changing the iteration trap: Replace an enumerator with an iterator. An implementation could provide an efficient way of converting one to the other. iterator: function () -> {next: function() -> any} Debate about why this is an object with a next field that contains a function rather than just a closure. Some rationale: this makes iterators more similar to generators that have several fields. Iterate trap returns an iteration driver. Lots of confusion about trying to understand the iterator API proposal, its terminology, and the code on the wiki. Can't usefully continue to discuss it until the confusion is cleared up. Question about performance implications of using exceptions to terminate iteration. On the other hand, it is more convenient. **Mark Miller**: Conflict between two levels of abstraction (say nested for-in iterations), where the outer one wants to throw a StopIteration exception through the inner one.

Brendan Eich: Mozilla experimenting with not suppressing deleted properties from enumeration. Instead, they'd make a snapshot and iterate through the snapshot. This is incompatible with ES5 as written but ought to work in practice and is much simpler than trying to track changes in the prototype hierarchy.

7.4 for-loop ambiguities

7.5 iterators

7.6 generators

7.7 shallow continuations

Shallow continuations: Not just a syntax change from the last proposal. The semantics are different: finally clauses will not get called if control escapes out of the function without calling the continuation.

Waldemar Horwat's objections: A function does something completely different if anywhere inside it there is a shift or yield expression. These non-function-like things should be distinguished (perhaps with a different keyword than "function"). Would be more amenable to this proposal if try/catch/finally were not allowed in outer scopes of such things.

7.8 simple modules

Simple Modules proposal: Banishing global object is done by the script tag, not by modules. module M = Math is similar to const M = Math but with eval-when :compile semantics. Controversy over the local renaming syntax: whether import x.{A:B, C:D} means A is defined with the value B or B is defined with the value A. Inner modules are exported by default.

Remote modules on the web (1) example: json2.js is a module body, not a module:

```
// json2.js
import stdlib.{String, ...}
var tmpPrivateThingy = ...
export function Stringify(x) {...}
module Submodule {export function Foo() ...}
```

Caller:

```
module JSON = load 'http://json.org/modules/json2.js';
```

```
alert(JSON.stringify({'hi': 'world'}));
JSON.Submodule.Foo(...);
```

Note that json2.js does not name itself at the top level; the caller gets to name it. **Allen Wirfs-Brock**: Some disagreement with that--the module should get to name itself in the default case. Doing that that way would complicate multiple modules in a file though.

Waldemar Horwat: How does the json2.js module refer to itself? Answer: it can't?

Unhygienic name capture concern. **Doug Crockford**: A widget needs to be able to defend itself from the page.

7.9 module loaders

Filesystem modules for offline ES example has a bunch of bugs. Top-level modules are superfluous in the example. Question: what happens if one writes var h = io.File.open(...) instead of var h = open(...)? If it works, then the file system hierarchy is overlaid over the variable environment--creating a file in the current directory will create a variable in the program. If it doesn't work, then import does a different lookup from the normal one.

Cyclic dependencies:
Can two script tags have cyclic dependencies? No.

```
// main file
module Even = load "even.js";
module Odd = load "odd.js";

// odd.js
import Even.*;
export function myOddMethod() ...
...

// even.js
foo = Odd.myOddMethod();
// Note that "Odd" is within scope here!
...
```

Mark Miller: Objects to resolving ModuleLoader.current() using dynamic scoping to get the module loader. Confusion reigns after a bit of discussion. What if the call got initiated by the module eval inside another module eval? Here it's controversial. What if it got initiated by an event handler? The model doesn't seem to work in this case yet. Do we need ModuleLoader.current()?

ModuleLoader uses its "global" parameter only for legacy code uses.

ModuleLoader's "load" parameter is expected to call exactly one of its three Request functions. It's an error if it doesn't call any of them. rewriteMRL never reinvokes some other user module loader; it only uses the system one.

evalScript throws an error if it gets an export statement. However, it permits local modules but does not export them.

Not clear yet how dynamic loading works with mutually recursive modules.

7.10 let expressions

Agreed to let expressions move to the graveyard. They don't carry much weight and seem like a grab bag feature.

7.11 typed arrays

Consensus that aliasing and exposing endianess are problems. Would be more interested in a solution that doesn't suffer from those.

Khronos group joint meeting: The issues are visible endianness and aliasing. Khronos found that the operations that they needed for file i/o were somewhat distinct from those they needed for graphics work.

Possible alternatives that don't expose endianness: Lightweight structs. Khronos: it's less efficient to index into those. Why?

Arrays of structs are preferable to sets of homogeneous element arrays due to cache and convenience effects.

Allen Wirfs-Brock: Trade off lots of different views of scalar values vs. A richer set of SSE-like operations that might be a better semantic match. Example: smalltalk bitblt allowed a rich set of customizations.

Don't currently have scatter/gather operations. Would like to have them.

Well-supported arrays of structs (as schemas) might be a satisfactory compromise. Consensus emerging around that solution.

Khronos would like to continue developing their API as WebGL host objects in the meantime. This may lead to forking of APIs in the developers' minds. The possible danger is failure to reuse common abstractions.

Need a champion. **Waldemar Horwat** offered to work with Khronos to drive a proposal.

7.12 proper tail calls

It's very hard to make any claims about space complexity or even reachability of tail calls. There's a research project which came up with a complicated framework for making such claims, but it's probably too much for our spec.

Even the following is not currently clearly defined: Function F has local variables X and Y. A closure capturing X escapes. F returns. Is Y live or not? Would need to define this first.

Allen: Tail calls would have slow uptake among implementors.

Tail calls are hard to do in Rhino.

No consensus on tail calls.

8 Any other business

8.1 Khronos

TC39 possible relationship with Khronos (WebGL WG and Typed Arrays):

The question was brought up if we should have a joint meeting with Khronos (WebGL WG) technical side. This was not clear on the first day, because of the heavy agenda of the TC39 meeting. A technical discussion on Typed Arrays was contemplated on the second day in the afternoon at 15:00 (see details in 7.11 above on Typed Arrays).

Mr. Arun Ranganathan [arun@mozilla.com] - the Chair of the Khronos WebGL Working Group - briefly introduced Khronos (http://www.khronos.org/) and the WebGL Group. He expressed their interest in working together with TC39 (They are interested in an open test suite).

From TC39 side (**Mr. Neumann** and **Mr. Sebestyen**) it was requested to get more information on the Khronos IPR policy. It was promised that such information would be made available to TC39. Verbally it was mentioned that Khronos specifications are free of charge to download, and also implementation of the specifications is Royalty Free (for patents). Khronos has also trademarks, and the use of those trademarks are regulated by internal policies.

Mr. Wirft-Brock could imagine a workshop with WebGL.

8.2 Decimal

Mr. Miller has reported that since the March 2010 TC39 meeting he did not have any interaction with **Mr. Ruby** from IBM.

Mr. Ruby:: Can decimal be done as a package? Yes, if doing it via function calls is sufficient. Replacing the behavior of "+" would be harder.

Mr. Miller: Value types not as important as other things that have already been dropped from ES6.

Mr. Crockford: Would want to replace double with decimal, but 754r is too flawed.

Mr. Miller: Disagree with Mr. Crockford about double but agree that 754r is a non-starter. There are two issues “value type” and the language. The “value type” is still an issue and we should persue it for the next Edition, but the language issue is less important (also in the light of the announced retirement of the respective IBM expert Mr. **Mike Cowlishaw**).

Mr. Neumann said that the fact that for about a year there is so little progress that is an indication to him that there is no interest in the topic. He observed that currently there is no “driver” behind this topic in TC39, neither by IBM nor by TC39 members. He was afraid that he cannot fulfil his promise to **Mr. Breidthardt** to push forward this project in TC39. **Mr. Neumann** would like to go on record very early on that as things look like right now “decimal” will not be part of the next Edition of ECMAScript, because no body is doing at the moment any serious work and apparently no interest. This issue will be brought up on day 2, when **Mr. Ruby** will be on the call. What is required for inclusion is that a strawman proposal needs to be posted on the Ecmascript wiki, followed by discussion, and a proposal. All this needs to be done within the coming year otherwise it will be too late. A deadline within the committee for making final decision on content for the next version of ES-Harmony will be June/July 2011 in order to complete all testing and work by the end of 2013.

9 Date and place of the next meeting(s)

July 28 – 29, 2010 Redmond, WA (Host: Microsoft)

September 29 – 30, 2010 Location TBD

November 1 – 5, Lyon France (with W3C) or 17 – 18 Location TBD.

10 Closure

The chairman reviewed open issues for the next meeting, indicated that a draft agenda for the next meeting would be circulated in the next couple of days and hope to produce agenda by end of next week.

W3C got a request from Google for an ECMAScript internationalization library. We should invite these folks to the next meeting.

The chair noted the above was received in an email from W3C and added it to the next agenda for discussion. The next meeting will be in Redmond, WA at the kind invitation of Microsoft.

Mr. Neumann closed the meeting at approximately 16:10. He thanked the delegates for their hard work, Google for their kind hospitality and Ecma International for the Social Event.