These notes cover: ECMA/TC39/00/3

ECMA TC39 meetings 13th July 2000

Orange County Convention Center, Orlando, Florida, hosted by Microsoft

Present:

Andrew, Sam, Herman, Dario, Waldemar, Jeff, Mike M., Chris, Ted, Lewis

Edition 4 Spec

Herman lead off with a meta-discussion, the issue being the perceived glacial pace and lack of closure on issues. His proposal was that if we find ourselves at point of controversy, we immediately look to strike the function and continue on.

new features (example: the meaning of square brackets in classes), then I don't know what "striking" means. Waldemar indicated that removing the consistency with objects defined by prototype was in his mind tantamount to removing classes.

Herman challenged everybody to find a concrete real-life usage of referencing predefined attributes via the square bracket notation.

Waldemar agreed that the [] operator should not provide access to private attributes. Herman argued that from a performance point of view, he wanted the meaning of the [] operator to be defined in a way that was independent of the caller's context. Getting access to the context and making decisions based on the result was simply unacceptable

He would only be willing to entertain such suggestions if people could identify a compelling benefit.

Chris noted that the implementation must provide a means to support a legacy mode, so the question couldn’t be one of development cost.

Herman argued that a full reflection machinery would be the right solution. I asked if it would be worth pursuing this for ten minutes or so in order to see if such a discussion would provide any insight into this issue. While Waldemar agreed that there was a need for a full reflection facility, he did not feel that even if such were defined, it would change his position on the necessity of defining the meaning of [] in a way that was consistent with the current standard.

By default, the [] operator will see public base version attributes of all new classes. It is

this behavior, as well as expando properties. Given that this would provide Microsoft customers a way to avoid this overhead, Herman noted his displeasure, but accepted the

general agreement.

---

expando properties. In option fast, it likely will not return any of these, but the behavior of Microsoft's option fast is outside the scope of the standard).

In cases where static type of an object is known at compile time, the current Microsoft

defined by that class. Waldemar points out that a subtype could have introduced such an attribute, so it cannot be an error.

Reluctantly, Herman agrees that this will be downgraded to a warning in the Microsoft implementation, and emitting a late bound implementation in such cases.

At this point, discussion went into what would happen if overloads were added to the language. Netscape and Microsoft have significant differences on how such a concept would interact with static types, so it was decided that overloads as well as static types (as opposed to type annotation) are not going to be part of the ES4 specification.

Herman bring up the point of whether types by default should include nulls; in Waldemar's current proposal, this is not the case. Waldemar agrees that this should be changed. Herman indicated that he would be open to some mechanism for defining "not null" as a precondition, but not as a part of the type calculus.

It was decided that this was not an essential feature, so it will not be part of this revision of the standard. Netscape is free to provide an extension that supports this and Herman indicated that Microsoft would not implement an incompatible extension.

Waldemar asked whether value types should include nulls. The general agreement was that value types should not. Waldemar asked whether user defined value types should be provided; and Herman indicated that it wouldn't be a bad thing - as long as value classes were final and did not exend other classes. This was deferred with the understanding that while it would be a good thing if a user defined value type proposal were added, it would not be fatal if this didn't make ES4.

A discussion concerning how ECMAScript should deal with subclassing of objects which are predefined by the host in a language which supports overloading. We decided that this was outside the scope of the language. Herman asked Chris for his opinion, and Chris indicated that he felt that the best map of this onto ECMAScript would be to map them onto a single method that does dynamic dispatch. Dario was uncomfortable with this recommendation.

Waldemar indicated a desire to unify "any" and "object". This has implications, in

would be:

```
true.x=5;
print true.x;
```

While this was valid in ES3, it was bizarre. What would happen is that "true" would be wrappered, and the wrappered object would then get an expando property added, then the wrappered object would be made available for garbage collection. Then "true" would get wrappered again, and this new object would be queried for an expando property. The general consensus was that this unification should be done, but Microsoft expressed a

in terms of the breaking backwards compatibility.

Herman brought up the topic of named parameters. The concept is to provide a small number of arguments in any order out of a large list which may have optional values. In Waldemar's proposal, named parameters become properties of the rest parameter. Proposed syntax:

```
obj.meth(arg1, arg2, name5:value, name4:value);
```

Waldemar agreed to update the proposal to conform to this syntax. Herman agreed that named parameters were a desirable but not essential feature of the language. At this point, we had agreement on the syntax of calling out to methods that support named parameters (for example, host objects), but not on how ECMAScript objects could be defined to accept named parameters.

Herman noted that in C#, rest parameters is an array. Herman indicated that it would make life simpler for him if ES4 were similar. Waldemar described a use case of a proxy object which wants to accept a rest object in order to pass it on. This was given as a reason for wanting to have named parameters for rest arguments.

There was no agreement on this, so until this is done the proposal is to defer the ability to

multiple incompatible extensions.

We agreed to come back to this one. The primary disagreement on whether the default should be that rest is an object or an array.

WAP Requirements

Ted went over objectives of WAP. He promised to send his slides to the mailing list.

Their stated goal was ECMAScript conformance - and review had been done to identify issues. In many cases, the WAP group will resolve these issues internally. The following was list of issues were ones identified as ones that they could use our help to resolve:

- Desire to replace WAP variables with EcmaScript variables. This certainly was met with approval.
- WMLScript currently uses "." for library calls. The suggestion is that WML treat

libraries as an object with static methods.

- WAP is going to publish a byte code definitions, and desires suggestions for standards. MSIL and JVM were provided as examples.
- What should be provided in terms of libraries? There was some discussion of having the WMLScript "string" type subclass and extend the EMCAScript string class.
- Pragmas - WAP desired the ability specify compilation unit level information. Example: access control.
- Numeric values - 32 bit integers and floating point. Strong concerns still remain about supporting 64 bit format numbers. It was noted that the existing ECMAScript implementations provide the external behavior specified in the specification without using double precision floating point in all cases internally. The suggestion was made that WAP applications should provide an emulation library, but do most of the math in such a way that it is not necessary. It was also suggested was that WAP consider standardizing on decimal. WAP requested rough sizings for emulation libraries.
- Script functions missing - essentially this is host objects. Not an issue. Question: should this be based on ES4 objects or packages? The working group suggested that it should. Ted suggested that the ES WorkingGroup should bump the priority of resolving issues related to "packages".
- Garbage collection - current spec is silent on memory management. WAP may request weak references, this was met with approval. Herman indicated that explicit memory management was not viewed as compatible with Ecmascript. Others agreed.
- Eval would be optional in WMLScript - removing it would be OK - WAP has apply.
- URL based function calling - compilation units can be named via URLs. We should consider a deferred import in ES.
- Versioning - WAP is interested in the syntax for versioning. This means that WAP would need to be based on V4, for classes, packages, and the like.
- Naming - WML has compilation units. Suggestion is to use E4 packages. It was requested that the WML group not take the E3 spec and extend it to support something that will compete with the work being done in E4.
- Schedule WAP wants responses back no later than the 7th of September, and WML will respond by the 15th. Final response desired Nov 10th.

CSharp discussion

At this point, Tony Goodhew and Jim Miller joined the meeting in order to give the ECMAScript committee an opportunity to ask questions about the C# and CLI proposals.

Q: Why did Microsoft propose organizing this effort as new subgroups of TC39 rather than pursue another organization?

A: Microsoft is has been working with TC39 for quite some time, and feels that the group works quite well. Compared to other standards bodies, it proceeds fairly rapidly, as evidenced by the fact that we are now working on the forth edition of ECMAScript. Additionally, the committee itself seems to respect the need for delivering working implementations and the value of

customer input.

Q: What value does Microsoft expect standardizing C# and the CLI would bring to Microsoft?

A: It certainly would encourage much broader participation. As an infrastructure, Microsoft is very interested in this being available as widely as possible.

Q: Does "as widely as possible" specifically include platforms such as Solaris, AIX, and Linux?

A: Yes it does - Microsoft is very interested in this happening. Jim expressed his private doubts that Microsoft itself would be the ones to implement versions for such platforms, citing both economic and political reasons.

Q: Will Microsoft be open sourcing their implementation?

A: This is under consideration, but has not been decided. Microsoft has been approached by a number of companies desiring to partner on this. Jim expressed his opinion that he saw it likely that the source to a reference implementation would be made available, but declined to speculate on the licensing details.

Q: Will this include the associated class libraries? Such as the ones needed to support the Python and Perl implementations?

A: This will likely include any classes necessary to support the standard. As far as the remaining classes, nothing has either been ruled in or ruled out. There are some classes which are platform specific, and Microsoft has made a first pass at splitting out the "system" classes from "microsoft" classes. Microsoft expressed an interest in working with committed in order to determine the rate at which these classes could be digested - suggesting that it was unlikely that the full set would become standardized in the first revision of these standards.

Q: Does Microsoft intend to release their implementation prior to the standard becoming adopted?

A: Microsoft plans to participate in the C# and CLI in much the same way that they have participated with EcmaScript - there may be times when functions will be implemented in anticipation of the standard with full preview of the working group. Should changes come about between implementation time and standard time; MS will work to conform. Depending on release schedules, conformance may not always take place at the release immediately following the publication of the standard.

Q: What are the relative priorities of the various languages that Microsoft has implemented on top of this CLI?

A: There are on the order of 15 languages being developed on top of the CLI at this time. Jim wasn't aware of any Microsoft assigned priority, but his observation was that C#, C++, VisualBasic, and ECMAscript (a.k.a. JScript) were certainly given focus.

Q: Have either of these proposals been reviewed with academic communities?

A: Both have been reviewed extensively with various academic communities. In fact, these communities had encouraged Microsoft to pursue standardizing this, and had expressed a hope that the standards body selected would welcome their continued participation. Andrew agreed to find out whether or not this would be possible. There was general agreement by the working group that this would be a good idea.

Q: Does Microsoft have any problems with these minutes being published?

A: No, as long as there is a distinction is made between answers representing Microsoft and those answers which were the opinion of an individual (example: language priority, above). Jim and Tony requested that their personal e-mail addressed not be posted.

Q: Will there be a newsgroup set up to discuss this?

A: Tony is working on this, and once one is available, he will provide the information to Andrew for distribution to the working group.

Q: What are the timeframes that Microsoft is anticipating for this to become a standard?

A: Microsoft plans to formally propose this at the next business meeting in Bristol. While it seems unlikely that the standards work could be complete by the end of this year, the hope was that

something could be produced next year. Microsoft, however, would prefer that dates be set by the committee rather than by Microsoft.

Q: To what extent is Microsoft willing to accept changes to these specifications?

A: As long as the changes are established via consensus and respect the impact to the existing tools and users, Microsoft is OK with change. This means that additions to the CLI would be less of a concern than non-backward compatible modifications; similarly changes to C# would be less of a concern than changes to the CLI.

Q: What is the status of ISO compliance of C++. Specifically of interest was the ability to use STL with _gc marked classes.

A: Microsoft is working towards ISO compliance, but was not able to comment on dates.

Participants

Andrew Clinick	Microsoft	andrewc@microsoft.com
Chris Dollin	HP	kers@hplb.hpl.hp.com
Jeff Dyer	Mountain View Compiler Company	jeff@compilercompany.com
Waldemar Horwat	Netscape	waldemar@netscape,com
Mike McCabe	Netscape	mccabe@netscape.com
Sam Ruby	IBM	rubys@us.ibm.com
Dario Russi	Microsoft	drussi@microsoft.com
Lewis Theran	Nokia	louis.theran@nokia.com
Herman Venter	Microsoft	hermanv@microsoft.com
Ted Wugofski	Phone.com	ted.wugofksi@corp.phone.com