Date: 13 March 1985
Attending: Norm, Eric, Hank, Andrew

Goals of Oz:

1. We claim that we can use a single model of computation (objects) everywhere, both for local data and distributed data. This can obviously be done inefficiently, we claim that it can be done efficiently.
2. One method of proving this is building a system that demonstates it. Such a system will not be allowed performance excuses.

It will be Norm's jobs to see that local invocations execute as fast as local procedure calls, and Eric's to make sure that remote invocations go faster than Eden.

Things that we are not concerned with:
  Checkpoint or other stable storage mechanisms.
  Alias checking - verification.

Things that we have to be concerned with:
  Invocation - parameters
  Language
  - as simple as possible but no simpler
  - elegant
  Concurrency
  - TOCS Aug 84 Synchronizing Shared Abstract Data Types
  Garbage Collection
  - It is clear that toto will require a garbage collector, not as clear that we will actually have to implement it.
  Location
  - Invocation is location independent, but objects are not.
  - Location dependent operations could be introduced as language primitives or as procedure calls on the run-time kernel. There is an analogy here with CEkernel calls. Things like CEwait, CEsignal, etc. are called as a result of programming language statements, while other things such as CEawait (for UNIX signals) are implemented as procedure calls to the kernel. The deciding issue according to Andrew is whether the location of objects has semantic repercussions. One that Eric and I came up with relates to failures. The failure semantics depend on location; things that are local will not fail for "Node crashed" reasons.

Implementation Approach:

We discussed a number of alternatives for implementing the system. These are:

- UNIX.
- VaxELN
- V-Kernel on Suns (or vaxen if they have it done)
- XEROX dandelions
- Eden
- SunOS

We discussed these alternatives at length. It was noted that as far as Norm is concerned, the choice of a system is not so important. These decisions have a far greater impact on Eric's development work. VaxELN, V, and SunOS have the advantage that they are small, that we should be able to get sources. This implies that we can use them to get the initial prototypes up quickly, and then change that parts that do not fit our needs, and finally cut away all the stuff that we don't need at all. The problems with Eden and Unix is that there is a large amount of stuff there, and that cutting away at it is not feasible. The XEROX environment has the additional disadvantage that we don't know anything about it, nor do we have any local experts. It was the conclusion that VaxELN and V offered the best possibilities.

Meetings:
We plan to meet three mornings a week. This week just 2. The next meeting will be Thursday 21 March at 10:00. Before then:

Eric:	Write a research proposal outlining the points that he wants to prove. This will include areas of interest and ideas for solutions.
Hank:	Look at VaxELN with the goal of deciding whether it is a good basis from which to start.
Norm:	Write to Guy Almes about access to V from Rice. Also write a research proposal.