

Synthesis - actor
AI tor
at memory transformations
actors
transforms co1

Thought Experiment

You have ∞ fast computer
 ∞ Storage ∞ fast

what do you do

A

- Stream audio, video, into memory
w/ persistent storage

- Set up graphics to display
a region of memory that
changes predictably every
frame

B

- Set up audio stream from
a predictable sequence of
memory locations

- Repeat A & B many times
w/ ability to give feedback
when something interesting
is observed

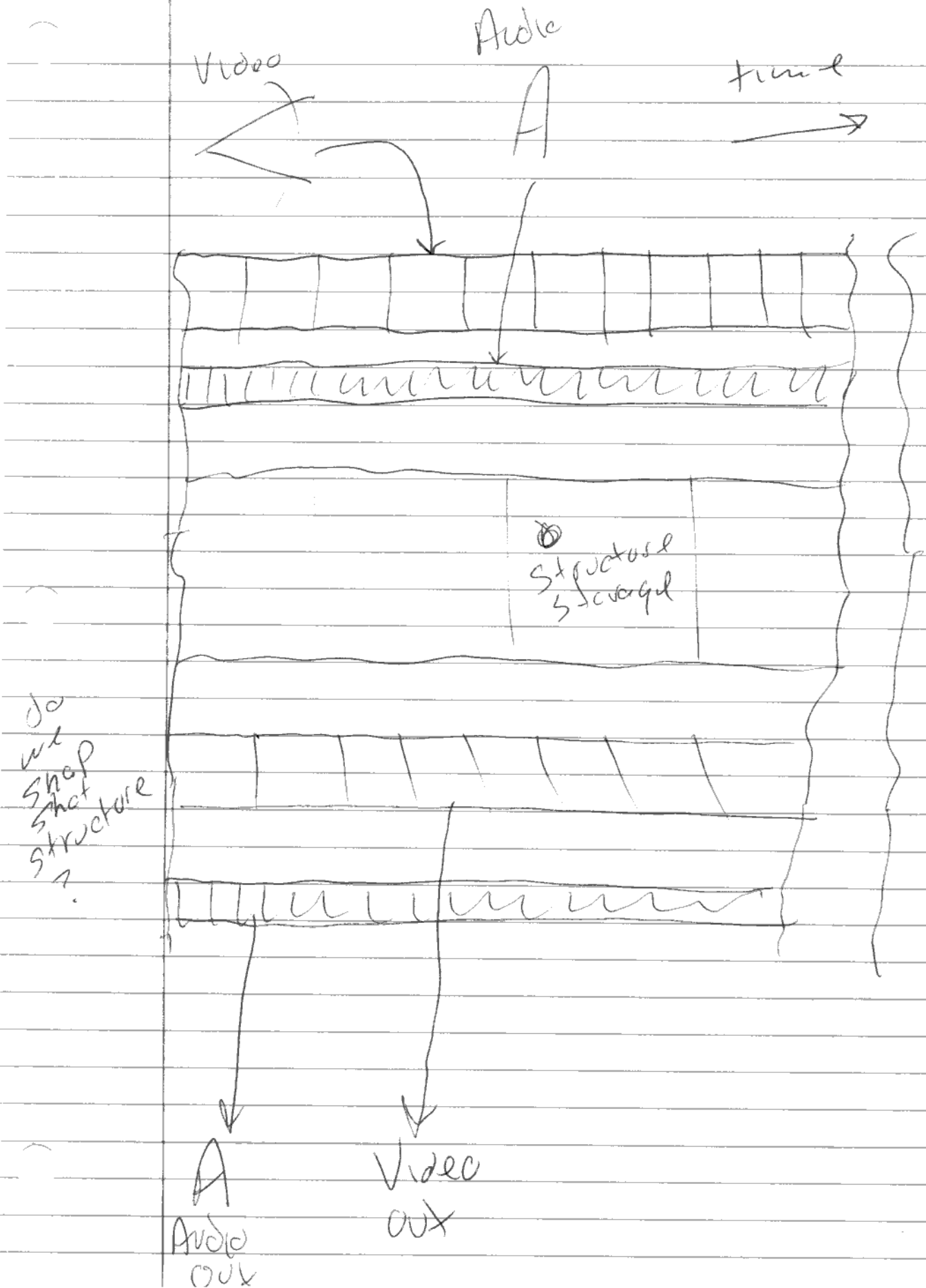
modularity

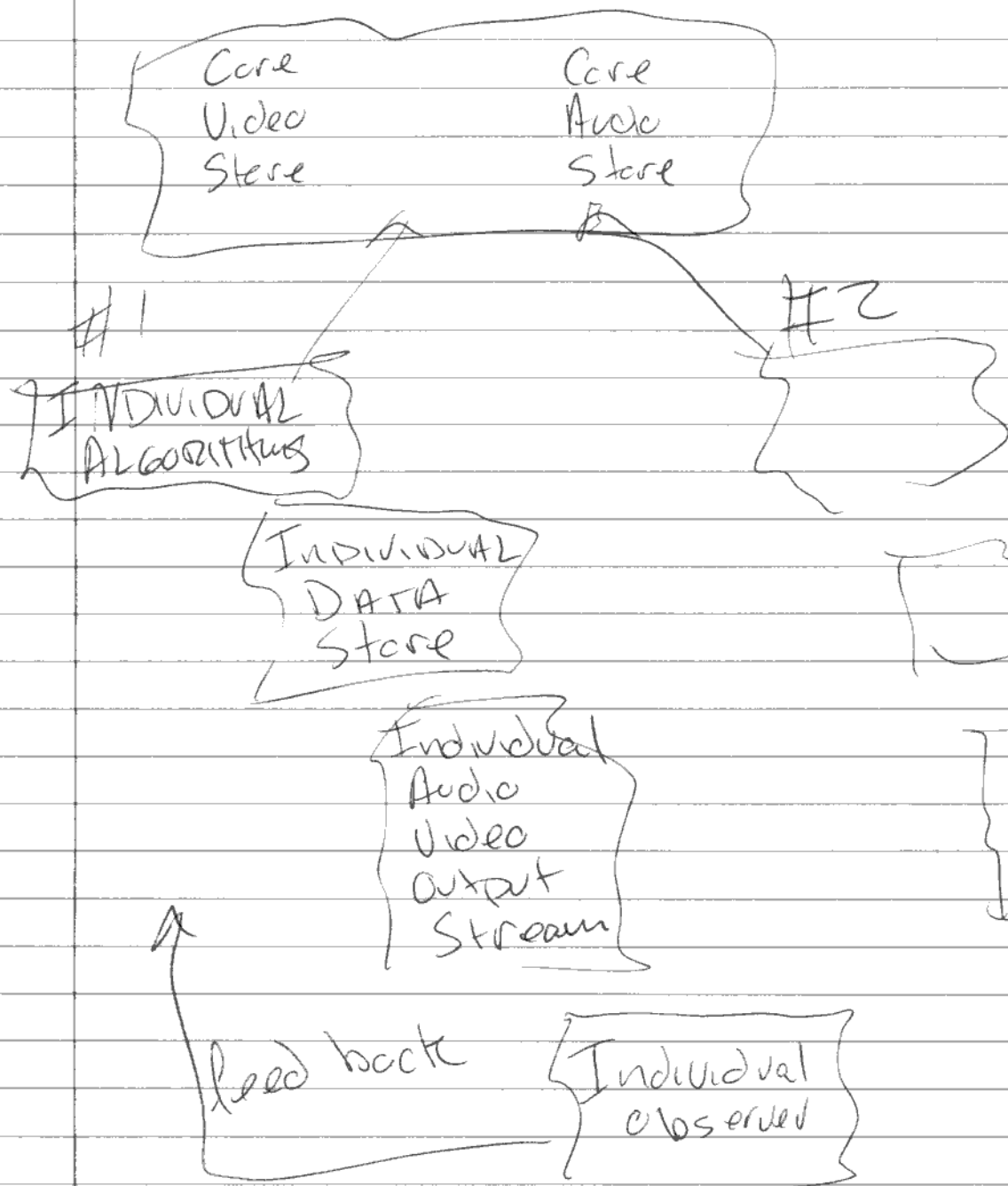
actors

self organizing

maximally decoupled

Full access to past



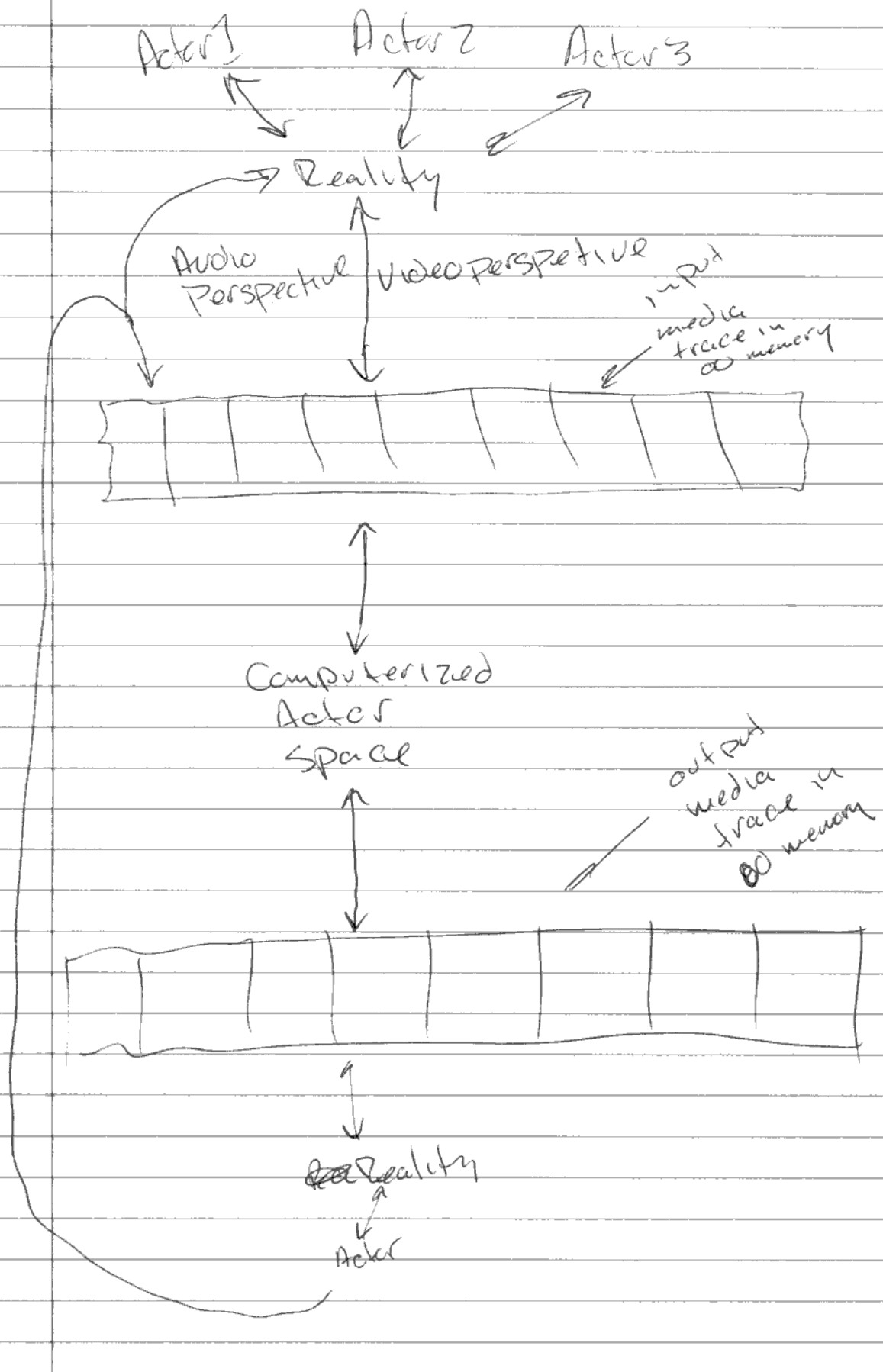


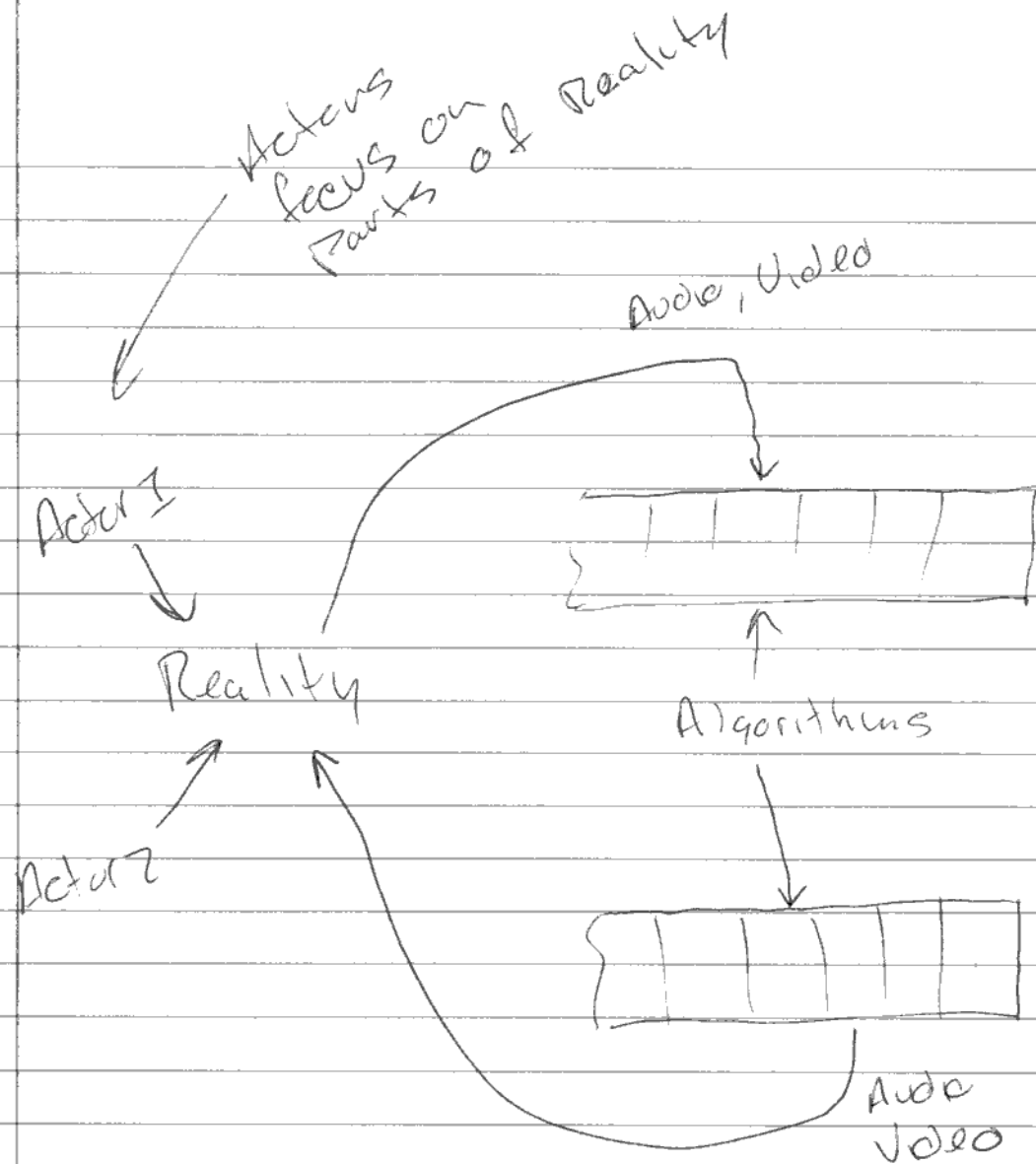
Requirement: ∞ Individual observers looking for the right or interesting algorithms.

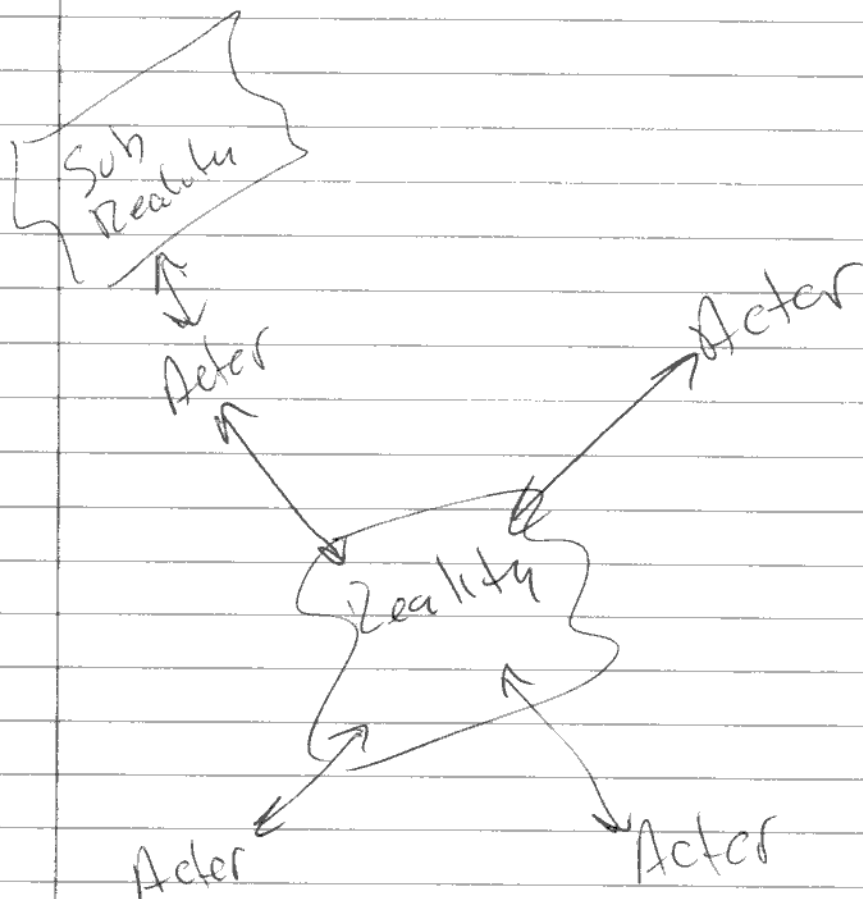
Another Problem

one of the individual observers
could find the best match
for the past but what
about the future?

Dont we want something interactive?



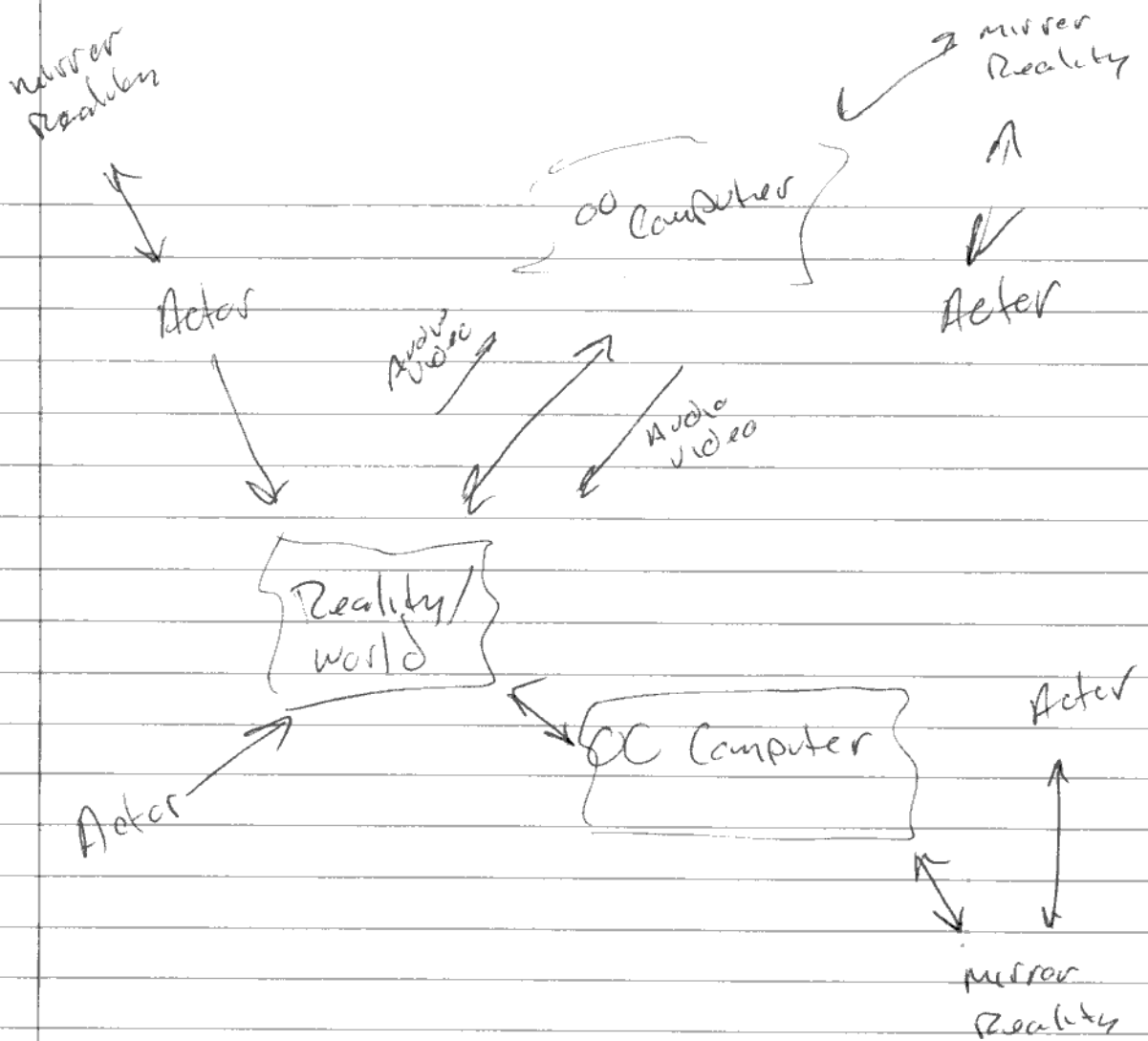


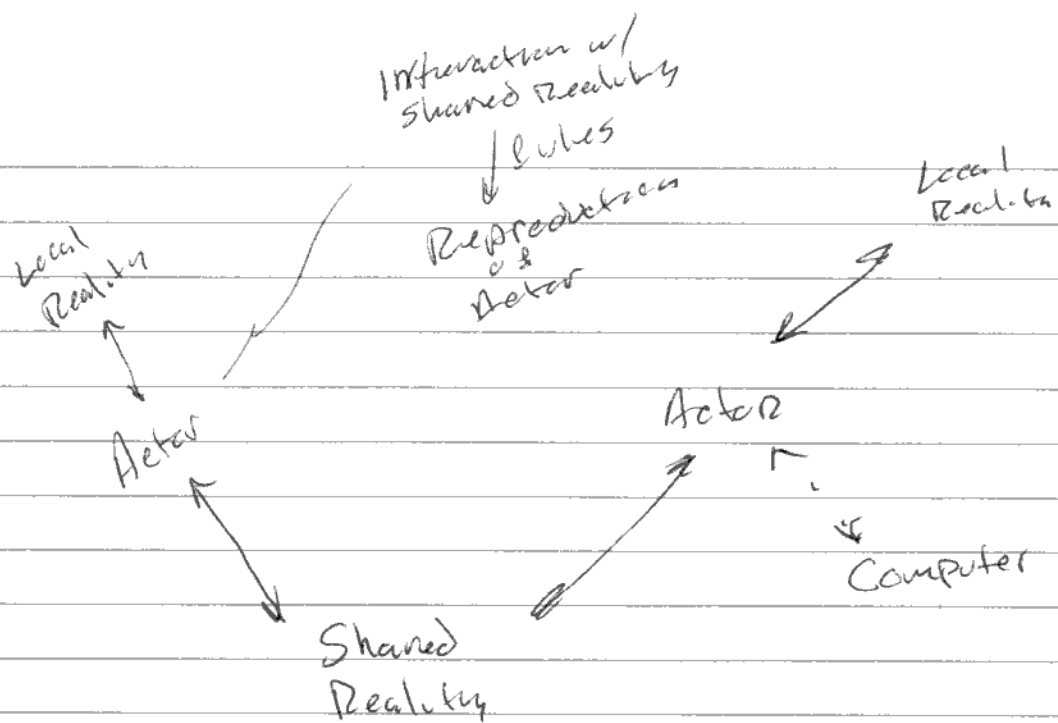


Reality is the common meeting Point

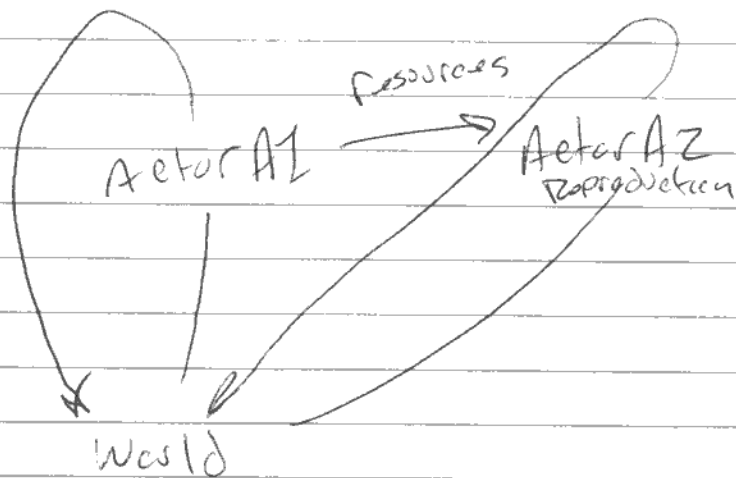
Add a computerized actors w/
a memory in each and
a compute / second in each and
a variety of algorithms

Add feedback of value of actors





- The level of manipulation of shared reality is important because the degree of success in that interaction determines the ability to reproduce the actor
- It could be argued that reality as we know it is the result of interaction of actors w/ various means of self propagation on a shared reality
- The computer as it exists today can only manipulate or more appropriately, influence the world through another actor. This limits the ability of useful actors or actions to propagate organically



This is the pattern

Is it dangerous to create a competing actor?

Important:

210		producers
000	0	Increasing discontinuities /
001	1	bits to infinity
010	2	Producers
011	3	
100	4	# enumerations \ll #bits
3bits		as each go to infinity

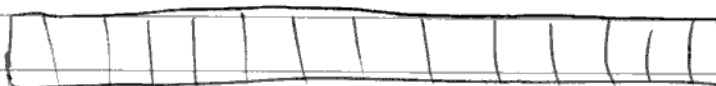
Fundamental limit on computability

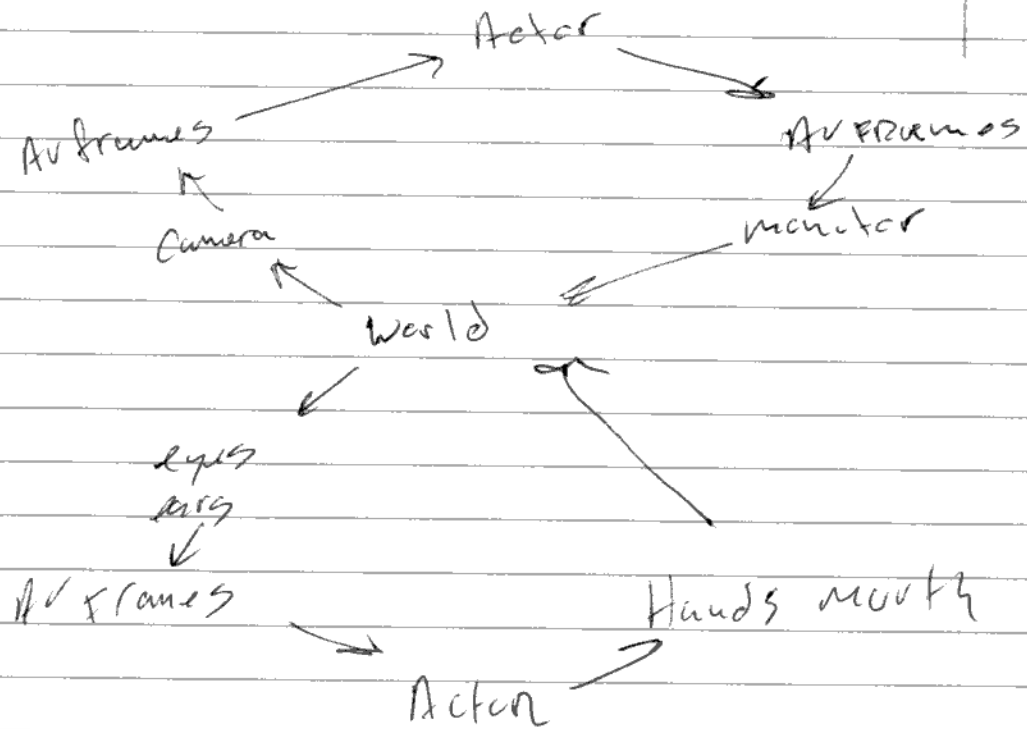
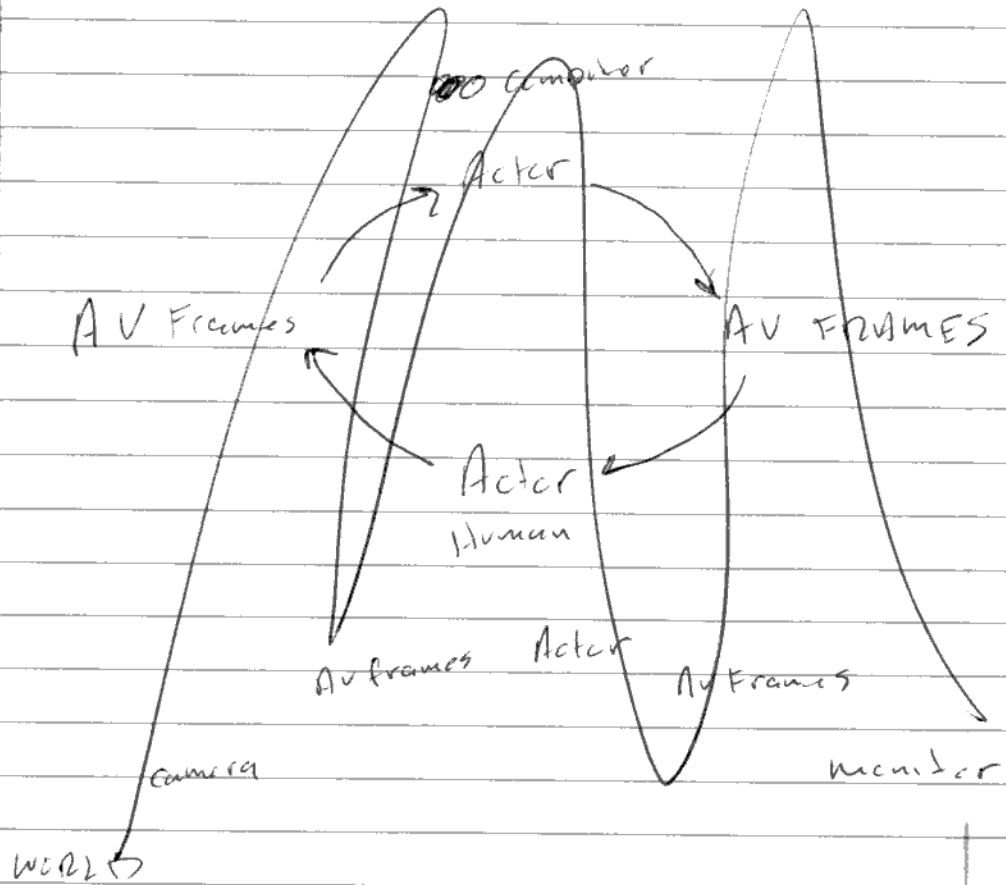
In
Keys/
Audio/
Video
Frames



OO Compute \Rightarrow Functions

Out
Audio/
Video
Frames





There is and must be a search function associated with any agent acting on the reality humans act on.

As such there is a fitness function applied to a result frame generated from a test of a hypothesis

As such the general economics of search apply in terms of the number of variables determining the size of the search space and search time and the profile of the fitness space being something that can be leveraged

As such there is a strong tendency to localized actors and larger numbers of those actors

A key factor here is not the limit of the compute power of the actor being evaluated but of the other actors acting on the world and driving the fitness evaluation.

Q) why parallel searchers for actors w/ limited scope?

Parallel

- Fitness Function is expensive to execute. if we prefer to test multiple Actors in single evaluation of Fitness Function
- in this case the fitness function includes waiting for the actions of other actors to provide feedback

Limited scope

- the search space explodes w/ # of variables.
- we want to hierarchically break down fundamental actors and build w/ these

Spemes like we could write formula to model this