PROFESSOR: [INAUDIBLE].

"Thus Spake Zarathustra" was made famous and popular by 2001.

And that is music played at this magic moment when some primate suddenly gets an idea, presumably one of our ancestors.

So how do we explain all that?

We've got all of the ingredients on the table.

And today I want to talk about various ways of putting those ingredients together.

So we talked about representations, we've talked about methods, and today we're going to talk about architectures.

And by the end of the class you'll know how to put one of those things together.

Actually, no one knows how to put one of those things together.

But what you will know is about some alternatives for putting those things together so as to make something that is arguably intelligent in the same way we are.

So that is our agenda for today.

We'll also talk a little bit more about stories.

I think it was in 2007 when the Estonians moved a war memorial from the center of Tallinn off to a Russian war cemetery.

Prior to that time the Estonians had been building up their national computer networks because they thought that computation was the wave of the future-- networks and all of that.

Shortly after the movement of that war memorial, someone brought the Estonian national network down-- a cyber attack.

It was widely believed to be the Russians.

There's a large Russian ethnic population in Estonia to start with.

And the movement of that war memorial irritated the Russians.

And so everybody everybody thinks that they did it.

But you know what?

No computer can understand the story I just told.

They can revel through all of the worldwide web finding information that's relevant to that, but no computer can understand the story I just told, except one.

You'll see a demonstration of that later on today.

So by the way, if you're interested in understanding the nature of intelligence, this is, of course, the most important lecture of the semester.

And I should tell you a little bit about what we're going to do on Wednesday.

Because for some reason the day before Thanksgiving tends to be a lecture that's lightly populated, except in this class.

Because I'm going to talk about the artificial intelligence business and what can be learned from it about how to avoid going broke when you start your company.

So for many of you that will be the most important lecture of the semester.

It all started back in the dawn age of artificial intelligence.

And really, it all started at Carnegie Mellon, sad to say.

Because the people at Carnegie Mellon, notably Newell and Simon, were the first to think about sort of a general purpose way of putting things together so as to build a structure or architecture in which particular intelligent systems could be built.

So their idea was called the general problem solver.

A long name for a simple idea.

And the simple idea is that you start your life out in a current state, call it C. And you want to get to some goal state.

Call it S. And a way you do that is you measure somehow the symbolic difference between where you are and where you want to be.

So that's the difference.

We'll call that difference D. And when you observe that difference that's enough, they say, in this general approach to problem solving.

For you to select some operation that will move you from your current state to some new state, an intermediate state.

Call it I. So I, or that operator, O, is determined by the difference, D.

And then, of course, the next thing to do is to measure the difference between that intermediate state and the state you want to be in, and choose some operator that's relevant to reducing that state.

So we'll call that D2, and we'll call this O2.

And D2 is what leads you to 02, and so it goes.

So that's the idea.

And that's often called means-ends analysis.

Why?

Because the end that you want to achieve is being in that final state, S. And the means is that operator, O. So you have some notion of where you want to be and the difference of where you are and where you want to be.

And you pick an operator so as to reduce that difference.

So this is all very abstract.

Let's exercise it in solving a problem that you will all be faced with here in a day or two.

That is, for many of you-- most Of you, I hope-- the problem of going home.

So here you are.

You're at MIT.

And where you want to be is over here, at home.

So you measure the difference between MIT and home.

And for many of you it's further than you can go by car and not so far that you can't go at all.

So what you do is, you say, well, the right operator is taking an airplane.

So there is the operator, take an airplane.

And this is the difference, D. And the difference, D, being sufficiently large, you take the plane.

Trouble is, if you happen to be sitting here in [? 10-250 ?] there's no way you can take an airplane, because they don't fit in here.

So you've got another problem, and that is to get to the airplane.

So the distance between here and Logan is such that the right way to do that is to take the MBTA.

And that's determined because you're working on this difference reduction right here, the difference from being at MIT and being at the airport.

So that difference dictates that you take the MBTA.

So you see, you're re-cursing.

But you know there are no MBTA cars in here either.

So there's still a difference like so.

And that difference dictates that you walk.

So you've got D1, D2, and D3.

And by the time you've excised the operators relevant to those three differences, you're at Logan.

Then you take the airplane, you get over to your hometown, and your faced with the smaller difference of getting from that airport to where you actually want to go.

So that's the general problem- solver idea.

It was such an exciting idea at the time.

It was such an exciting idea at the time because people would say to themselves, ah!

This is a general purpose problem solver, so we can set it onto the problem of making itself smarter.

And so there was a kind of imagined chain reaction that would take place.

And the developers of this architecture warned the public that within 10 years-- that is to say, by about 1970-- computers would be generally as smart as people.

And a lot of people made fun of them for that prediction.

But it was actually scientists attempting to be responsible.

Because they thought something, a quite serious dislocation was coming along, and that people should know that it was coming.

And so they felt it was their responsibility in that age of scientific responsibility to warn the public.

It didn't turn out that way, because the problem of collecting the differences and finding the operators, that's outside the scope of the architecture.

So this is the problem that has to be solved by a human before this architecture can be used.

You have to have identified the differences that you might encounter and the operators that you might use, and build this table which relates the two together.

So maybe that one, that one, some off-diagonal elements, and so on.

But building that table turned out to be a hard job.

So not surprisingly, the idea evolved.

And eventually the folks at Carnegie who developed the general problem solver-- most notably Newell and his students-- developed a newer, fresher, more elaborate architecture called SOAR.

And here's how SOAR works.

First of all, what does SOAR mean?

It doesn't mean anything.

It used to mean State Operator And Result.

But for some reason the proponents of the SOAR architecture decided they don't like that acronym, and have asserted that SOAR is merely a label that shouldn't be thought of as an acronym.

In any event, SOAR consists of various parts.

It has a long-term memory.

It has a short-term memory.

And it has connections to the outside world, maybe a vision system and an action system.

But most of the activity of the SOAR problem-solving architecture takes place in a short-term memory.

So you can view the contents of the long-term memory as shuttling in and out of short-term memory.

So you can see right away that this mechanism, this architecture, is heavily influenced by certain cognitive psychology experiments having to do with how much you can hold in your short-term memory-- nonsense syllables and all that sort of thing that was popular back in those days.

So this was an architecture devised primarily by psychologists.

And it had amongst its features a short-term memory and a long-term memory.

So that's part 1 of this architecture.

So what's in the long-term memory?

Well, assertions and rules, AKA productions.

A production being the Carnegie vernacular for rule.

It's just the rule-based stuff like you saw on almost the first day of class.

So the whole thing is a gigantic rule-based system with assertions and rules the shuttle back and forth from long-term memory into short-term memory where processing takes place.

The third thing that comes to mind when you think of SOAR architecture is they had an elaborate preference system.

You recall that when we talked about rule-based systems there's always a question of what do you do when more than one rule would work?

You have to have some way of breaking those ties.

The SOAR architecture has an elaborate subsystem for doing that.

But I said that these are the first three things you think, and maybe that's not right.

Because the next thing you think about is perhaps a better thing to identify with the SOAR architecture.

And that's the idea of problem spaces.

And that's the idea that if you're going to solve a problem you have to develop a space and do a search through that space.

Just like we did when we talked about how we can get from here to home.

There's a space of places, that's our problem space.

We can do a search through that space to find a way to get from one place to another.

That's the sort of thing that SOAR is focused on.

Finally, the fifth element that you tend to think about when you think about SOAR is the idea of universal subgoaling.

And that's the idea that whenever you can't think of want to do next, that becomes your next problem that deserves it's own problem space and its own set of differences and operators, and rules and assertions.

So you start off on a high level, then you have to solve problems at a lower level, just like you did up there with a general problem solver.

So if you have these two architectures you can begin to say, well, what are they centered on?

And this architecture, this general problem solver, is centered on the idea that everything is about problem solving.

Because the problem solving hypothesis-- no one gave it that name.

But that's what it was.

And this architecture did get its name.

And it was said always, by Newell, to be based on what he called the symbol system hypothesis.

The hypothesis that what we are as humans is symbol manipulators.

And we can uncover how that all works by giving people crypto-arithmetic problems and having them talk out loud, by thinking about what happens when you try to remember nonsense syllables, by all that sort of stuff that was en vogue in terms of psychology experiments in the day when this architecture was first articulated.

But when you look at architectures you can sort of see where they come from and what their antecedents are.

It has a short-term memory and a long-term memory, because Newell and his associates were cognitive scientists.

It has assertions and rules and preferences, because Newell and his associates were also AI people.

And it has problem spaces and universal subgoaling because those are ideas that had been work out in a more primitive form already in the general problem-solver architecture.

So that's a glimpse of what SOAR looked like in its early days.

It's been very highly developed by a lot of smart people.

So although it's symbol centered, they've attached to it things having to do with emotion and perception, but generally with the view that the first thing to do when faced with this perception is to get it out of there and get it into a symbolic form.

That's sort of the bias that the architecture comes with.

So those are two architectures that are heavily biased toward thinking that the important part of what we do is problem solving.

But the most important, perhaps-- at least from an MIT perspective-- of these problem-solving oriented ways of thinking about the world, is Marvin Minsky's architecture, which he articulates in his book "The Emotion Machine." And Marvin is not just concerned with problem solving, but also with how problem solving might come in layers.

So let me show you an example of the sort of problem what motivates some of Marvin's thinking.

So you can read that, it's a short little vignette.

You have no trouble understanding it, right?

No.

It's not difficult for us humans.

Awfully tough for a computer.

In part, because the thinking you need, your ability to understand that story, requires you to think on many levels at the same time.

First of all, there's a sort of, at the bottom, instinctive reaction.

You see where there's instinctive reaction?

That's the part where she hears a sound and terns her head.

That's instinct, right?

That's practically built in.

But then what she sees is a car.

So that's something that we don't have wired in.

It would be unlikely that we've evolved in the last 100 years to have an instinctive appreciation of cars barrelling down the road.

So the next level in Marvin's architecture is learned reaction.

So that's the part about thinking about the car.

Now spread throughout there-- well, let's see where is a particularly good example.

She decides to sprint across the road.

So that's where she's solving a problem.

So that's the deliberative thinking level.

It doesn't stop there, because later on she reflects on her impulsive decision.

So she thinks not only about stuff that's happening out there in the world, but she also thinks about stuff that's going on in here.

So that's a level which we can call reflective thinking.

Well, you know, it doesn't stop there, because she also considers, in another part of the story, something about being uneasy about arriving late.

So she's not only just thinking about events that are going on in her mind right now, but events that are going on right now relative to plans she's made.

Some Marvin calls that the self-reflecting layer.

But that isn't the whole thing either, because toward the end of the story she starts to worry about what her friends would think of her.

So there's a kind of reflective thinking in a more social context.

So he calls that self-conscious thinking.

So as the Carnegie folks think, the SOAR architecture focuses mostly on problem solving, Minsky's "Emotion Machine" book considers not just thinking, but thinking on many layers.

And the blocker to doing any of that can be said to be the development of common sense, which computers, alas, have never had much of.

So this could be said to be based on the common sense hypothesis.

And the common sense hypothesis holds that in order to do all of that stuff, you have to have common sense like people.

And if you have to have common sense like people, you have to think about how much of that is there and how can we go get it?

And so this spawned a lot of activity in the media lab amongst people influenced by Marvin, having to do with gathering common sense.

The open mind project, the work of Henry Lieberman and others, having to do with the gathering of common sense from the world wide web as a way of populating systems that would lay the foundation for doing this kind of layered thinking.

So that is a brief survey of some mechanisms, some older than others, but all but GPS-- GPS too.

Let's face it, it's hard to think of solving any problem without means-ends analysis being involved.

So GPS isn't wrong, it's just not the only tool you need to think about what to do.

So these are early, and late, and still-current.

But it's not the only thing there is, because there have been reactions against this problem-solving way of thinking about the development of intelligence.

And the most prominent of those counter currents, of those alternative ideas, belongs to Rod Brooks and his subsumption architecture.

So along about the early-- along about the years surrounding 1990, Brooks became upset-- subsumption-- because robots couldn't do much.

They would turn them on at night, and then the next morning they'd come in the laboratory and they would have moved 25 feet, nicely avoiding a table perhaps.

Not doing very much and taking a long time to do it.

So he had decided that it's because people were thinking in the wrong way.

In those days people thought that the way you build a robot is you build a vision system, and then you build a reasoning system, and then you build an action system.

And it can do almost nothing, but it does something.

So you improve the vision system, and improve the reasoning system, and improve the actual system.

And now you've broken it, because all the stuff you used to be able to do doesn't work anymore.

So what's the alternative?

Well, the alternative, as articulated by Brooks, is to turn this idea on its side.

So instead of having an encapsulated vision system, an encapsulated reasoning system, and an encapsulated action system, what you have is layers that are focused not on the sensing and the reasoning and the action, but layers that are specialized to dealing with the world.

So in Brook's way of thinking about things, at the lowest level you might have a system that's capable of-- well, before we get to that, avoiding objects.

And maybe the next level up is the wandering layer.

And maybe the next level up after that is explore.

And maybe the next level up after that is seek.

Now in the old days when people took 6001 I had no trouble getting an answer the question, what does this remind you of in 6001?

It doesn't remind you of anything in 6001 since you haven't taken it.

But it viewed, as a generalization of a programming idea, what is the programming idea?

There are only a few powerful ideas in program, and this is a generalization of one of them.

What is it?

Do you have a name?

Yes, Andrew?

STUDENT: Layers of abstraction?

PROFESSOR: Layers of abstraction, and abstraction barriers.

That nails it pretty well.

Because each of these guys can have its own vision, action, and reasoning system.

And if you think of these as abstraction boundaries, then when you got this thing working you don't screw with it anymore.

You build this layer on top.

And it may reach down in here from time to time, but it doesn't fundamentally change it.

Brooks was inspired in part by the way our brains are constructed.

All that old stuff that we share with pigs is down in there deep, and we put the neocortex over it.

So it looks layered in a way that would make [? Gerry Sussman ?] proud.

So this then is the way that Brooks looks at the world, and it's characterized by a few features just like SOAR is.

One of those features is no representation.

So this is a detail that's probably right at the level that Brooks was operating, and very questionable when you get above the level that Brooks was operating.

But before I go on, let me say what the hypothesis is.

The hypothesis is the creature hypothesis.

It's the hypothesis that once you can get a machine to act as smart as an insect, then the rest will be easy.

Well, how do you get a creature to be smart as an insect?

Maybe you don't need representation.

We focused on representation in this course, so you can see there's a little stress--- Next thing is, what do you do if you don't have a representation?

Let's see.

Your representation makes a model possible.

Models make it possible to predict, to understand, to explain, and to control.

So if you don't have one what can you possibly do?

Brooks' answer is, you use the world instead of a model.

So everything you do is reactive.

You don't have anything in your head that is a map of this room.

But maybe I don't need one because I can get around that table by constantly observing it.

And we don't have to fill up the memory with that information, I can just react to it.

So no representation, use the world instead of a model, and the mechanisms in their purest form are just finite-state machines.

So with that, Brooks was able to do things that people were never able to do before.

And what's the modern [? instantiation ?] of this architecture?

Now, according to Brooks, in use in 5 million homes in the United States?

STUDENT: The Roomba?

PROFESSOR: It's the Roomba The Roomba robot is, by Brooks' account, approximately the thirteenth business plan of iRobot.

And it's the one that made it big, because the Rumba vacuum cleaner has been very successful.

Would you like to see a movie of its processor?

So this is a film made some time ago that shows, in some sense, the summa of that architecture.

What I want you to imagine very briefly is a robot that wanders around in the halls and rooms of the old [? Tech Square ?] clinking the Coke cans.

Okay, you all got an image of that in your mind?

Because I want you to compare the image you now have of that robot that's wandering around collecting the Coke can, with the actual movie.

[VIDEO PLAYBACK] -Herbert, the soda-can collecting mobile robot.

He was built at the MIT AI lab in 1989.

Work was done by John Cannell under the supervision of Rodney Brooks.

Herbert is a robot controlled by subsumption architecture.

This is a collection of small behaviors that influence the overall activities of the robot.

There are no centralized controllers and no world model.

-Herbert navigates by using a number of infrared proximity censors around its body and basically following walls and corridors.

It can also look for the can through a laser light striper.

Right now it's come out of the door of an office, followed along the wall, and then its laser light striper has seen a can on top of the desk in front of it.

When this happens the robots and deploys its arm.

You can see the arm going out now.

-The arm has a number of censors itself.

There are fingertip censors, a break beam in the jaws, and two infrared proximity sensors on the front of the hand.

-It grabs cans in a stereotype fashion.

First, it lowers down to find a surface somewhere, then it bounces along the surface until it sees the can in front.

It uses the hand-based IRs to re-center the arm by rotating the robot's body until the can comes between the jaws of the gripper, at which point the break-beam senses the can.

-After acquiring the can, Herbert will have tucked the arm back into its normal traveling configuration and attempt to go home.

-Since it has no central [? arm presentation, ?] it doesn't have any map of where it came from.

Instead, it has an algorithm which uses a magnetic compass to determine every time it comes through a door, will it be able to find the door?

It basically has a policy of always going north every time it exits the door.

-So now the can is being tucked away.

As the robot turns you'll see a red stripe from the laser range finder.

And now it's using the [INAUDIBLE] IR to navigate back, find the door, and go through the door with its prize.

[END VIDEO PLAYBACK] PROFESSOR: And there, if you were paying attention, you saw a little glimpse of John Cannell who was the student to develop that system.

So that was a tour de force.

That was a magic moment.

That was when you open the champagne.

It's not what you expected, of course, because when I say imagine a robot wandering around in [? Tech Square ?] picking up Coke cans, that leaves open a huge envelope of possible hallucinations.

And usually or hallucinations about these things are-- we imagine things to be more fluid, more natural, and more impressive than they actually are.

But that was impressive, because no robot came close to doing anything like that before.

More to be said about that during the business lecture on Wednesday.

So that's the subsumption architecture.

By the way, maybe at this point we can say something about how the other architectures relate to what Minsky was talking about.

What's this deliberative thinking layer correspond to?

That's what SOAR is about, and maybe GPS.

So what's subsumption about?

It's about stuff down here.

It's about instinctive reaction and learned reaction.

But shoot, what about Minsky's other layers?

If we're going to be building systems that are as smart as those things then we have to worry a little bit about that sort of thing too.

So that brings us to the genesis architecture.

And now let me give you the standard caution that should be early in the presentation of any academic.

I will sometimes say "I," and what I mean is "we." And sometimes I'll say "we," and what I mean is "they." This was a system that was developed mostly by students of mine who persuaded me, after a great deal of time, that they were thinking the right kinds of thoughts.

But here's how the genesis architecture works.

As no surprise, given recent discussions, it's all centered on language.

And the language part of the genesis system has two roles, one of which is to guide, and marshal, and interact with the perceptual systems.

And the other is to enable the description of events.

That's how it works.

So is perception important?

I don't know.

I might ask you a question like, is there anybody sitting in the front row wearing blue jeans?

And it's hard for you to resist, under those circumstances, your eyes from going over there and answering the question.

Your eyes answer the question.

No symbol processing system is involved, except in so far as my language system has communicated with their language system, which drives your motor system and your vision system to go over there and answer the question for you.

But it's not just the real stuff that the language system directs your attention to.

It's also the imagined stuff.

It's been a long semester.

Have I told you the story about my table saw?

Probably not.

Here's the deal.

I bought a table saw.

It's a wonderful table saw.

I was installing it with a friend of mine who's a cabinet maker.

He said, never wear gloves when you operate the saw.

"Why?" I said.

Before he could answer the question I figured it out.

Can you figure out why you never wear gloves when you operate a table saw?

You know what a table saw is, right?

It's a table with a spinning blade in the middle.

And you use it to cut wood.

Why should you never wear gloves?

Yes?

STUDENT: Well-- STUDENT: --Well, you know the answer.

Ha, that's not fair.

That's old Brett up there.

He's heard the story too many times.

Yes, Andrew, you got it.

STUDENT: I've been told the answer before.

PROFESSOR: You've been told the answer.

How about somebody who hasn't been told the answer.

Yes?

STUDENT: Because the gloves might get caught.

PROFESSOR: Because the glove might get caught and pull your hand into the blade.

And then what happens?

It's horrible.

You're hand gets mangled and your fingers get cut off, and this happens a lot to professionals.

It won't actually happen with that table saw that I bought, because its play detects flesh and stops the blade, which then leads to stopping the blade and having the blade retreat into the table in about two microseconds-- two milliseconds.

So, in general though, it's a bad idea, and you always have to suppose that the mechanism isn't working anyway in order to use good safety practice.

But here's an example of something that nobody ever told you that he was able to figure out, by imagining what would happen and reading the answers off of the scene that he imagined.

So nobody ever says many of the things that we know, but we know them anyway.

Here's another example.

Imagine running down the street with a full bucket of water.

What happens?

The water splashes out and gets your leg wet, right?

You won't find that in Open Minds database.

Nobody ever said that over the web.

It's not written down anywhere.

But you know it.

Because you, we human beings have the capacity to imagine perceptual things and read the answers to questions off of our imaginations with that perceptual apparatus.

So that's a very important connection down there.

And then if you've got the ability to describe events, then you've got the ability to tell and understand stories.

And if you can do that, then you can start to get a handle on culture, both macro and micro.

And by macro culture I mean the country you grew up in, the religion you grew up with.

And by micro I mean your family and personal experience, and all shades in between.

So I don't know, what inspires me and my associates to think in these terms?

We talked about a little bit of it last time when I talked about evolution and the apparent flowering of our species about 50,000 years ago, at which time we got something.

And I believe that what we got-- and this is the characterization of this particular hypothesis-- what we got is the ability to tell stories and understand them.

So if we want to label this representation, it's the label strong story hypothesis.

So what's the weak story hypothesis?

The weak story hypothesis is, this is important.

The strong story hypothesis is, this is all there is.

But is there any other evidence of this is really, really, really important?

So I've queried Krishna here before the class starts, and he tells me I haven't told you about the following experiment.

This, in my way of thinking, is the most important series of experiments ever done in cognitive psychology, developmental psychology, actually.

So here's how we get started.

There's a rectangular room, if you're a person.

If you're a rat, it's a rectangular box.

All the walls are painted white.

Are you with me so far?

So now, in each corner there's a basket, or cloth, or something in which or under which you can put some food.

Now, you put the food there while the rat watches you.

And then you give the rat a little spin to disorient it.

All right?

So then, the rest stops and goes for the food.

And you can keep track of where the rat goes.

And the rat goes with approximate equal probability predominantly to those two corners.

So I'd have bet you didn't know that rats were that smart.

So they understand the rectangular nature of the room and they don't go to the diagonal corners where the food cannot be.

So are these genius rats?

Or maybe we're just rats with big brains.

Because we do the same thing.

So if you repeat this experiment and replace the rat with a small child, and then you put a toy in there instead of food, and the rat-- not the rat.

The child is usually held in a parent's arms, usually the child's mother-- usually because they think that if they participate in these experiments up there at Harvard their kid will get into Harvard some day.

So the kid goes to a diagonal corner just like a rat.

And then the next thing you do is, you try an adult, maybe an MIT student.

That way you can use food again.

And you get the same result.

Who could be surprised?

So rats, children, and human adults, pretty much all the same with respect to this experiment, until you paint one wall blue.

Rats are not colorblind, in case you're wondering.

Then what happens?

Well, if you pay one wall blue the rat still goes with equal probability to the two diagonal corners.

If you paint one wall blue, the child still goes to the two diagonal corners with approximate equal probability.

It's only us genius human adults who go only to that corner.

So this invites a couple of questions.

One of which is, when does a child become an adult?

Any ideas?

[INAUDIBLE], what do you think?

STUDENT: [INAUDIBLE].

PROFESSOR: You can pick a number greater than 1 and less than 10.

[INAUDIBLE], what do you think?

STUDENT: Five?

PROFESSOR: It's a pretty good guess.

Do you have siblings at that age?

It's a surprise but, why is it five?

Is it because-- what does it relate to?

Is there any correlate to the onset of that ability?

You might try everything, as [INAUDIBLE] does, because she's extremely careful.

So she's tried gender, she's tried the onset of language, the appreciation of music, handedness, and there's only one thing that matters.

And that is that the child becomes adult at that time when they start to use the word left and right when they describe the world.

Now I said that very carefully because they understand left and right at an earlier age, but they only have started to use the words left and right when they describe the world at the time that they begin to break this symmetry and go to the correct corner.

Now for the next element of this I need something to read.

Has anyone got a textbook handy?

Ah, "China, an Illustrated History." Now I need a volunteer.

OK.

Andrew, you want to do this?

So here's what you're going to do.

You can stay there.

But you need to stand up.

So what I'm going to do is, I'm going to read you a passage from this book.

And I want you to say it back to me at the same time I read it.

It's as if you're doing simultaneous translation, except it's English to English.

This things got words I can't pronounce.

OK, are you ready to go?

All right.

"When overwhelmed by the magnitude of the problems he tackled, he began to suspect that others were plotting against him or secretly ridiculing him." Thank you very much.

That's great.

So you see, he could do it.

Some people can't do it.

At least it take a little practice.

But he did it.

And guess what I've done to him?

I've reduced his intelligence to that of a rat.

Because if you do this experiment with an adult human who's doing this simultaneous English to English translation, they go with equal probability to the two corners.

So what's happened?

What's happened is you've jambed their language processor.

And when their language processor is jambed they can't put the blue wall together with the rectangular shape.

So it seems to be that language is the mediator of exactly the combinators you need in order to build descriptions.

Because they can't even put those things together when their language processor is jambed by the simultaneous translation phenomenon.

So that brings us to the two gold star ideas of the day.

One is, if you want to make yourself smarter you want to do those things-- look, listen, draw, and talk.

Because those are the particular mechanisms that surround this area down here, which is the center of what we do-- which is the center of our thinking.

So why do you take notes in class?

Now because you'll ever look at them before, but because it forces the engagement of your linguistic-- of your linguistic, your motor, and your visual apparatus.

And that makes you smarter, because it's exercising that stuff.

The second thing you can say, in conclusion, especially from this experiment, is beware of fast talkers.

Why do you want to be aware of fast talkers.

It's not because they will talk you into anything.

It's because that when they talk fast they're jambing your language processor and you can't thing.

That's why you want to be aware of fast talkers.

Because if they jamb your language processor you won't thinking and you'll buy that car, or you'll buy that drink, or you'll do any manner of things that people who want you to do those things have learned to do by talking to jamb your processor.

So that completes what we're going to do today.

And I'll give you a demonstration of some of this stuff on another occasion.