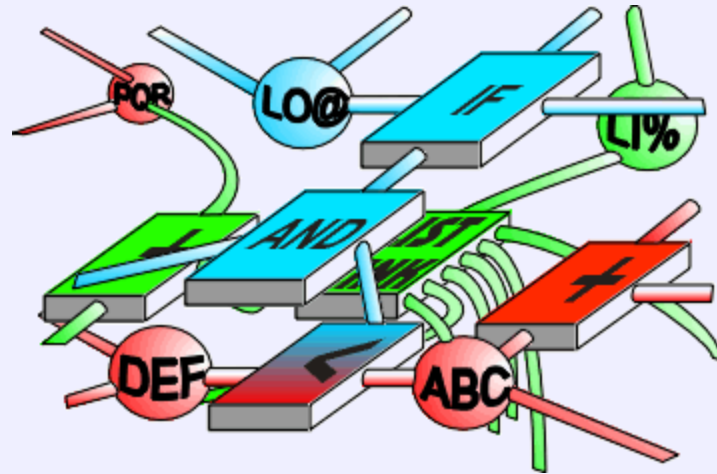
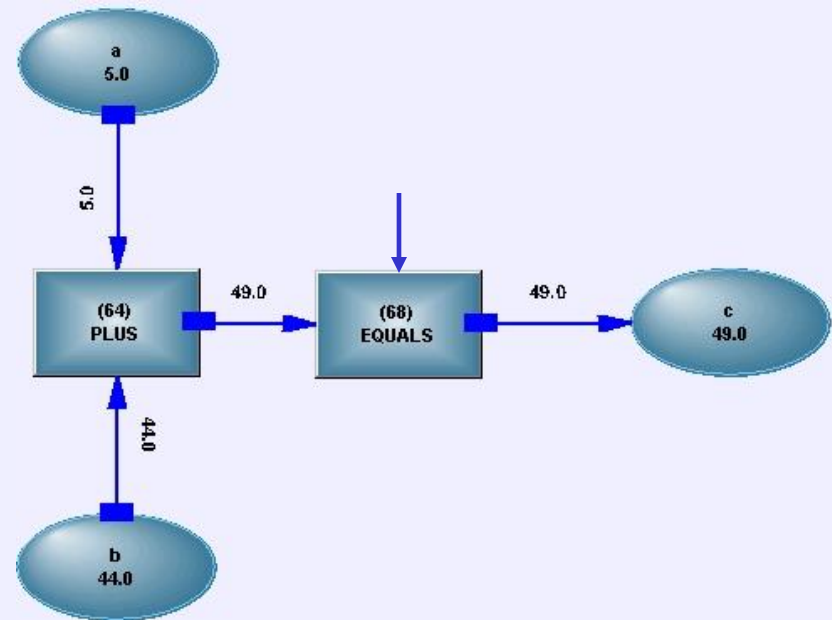


# Knowledge Networks or “The Rise of the Machines”

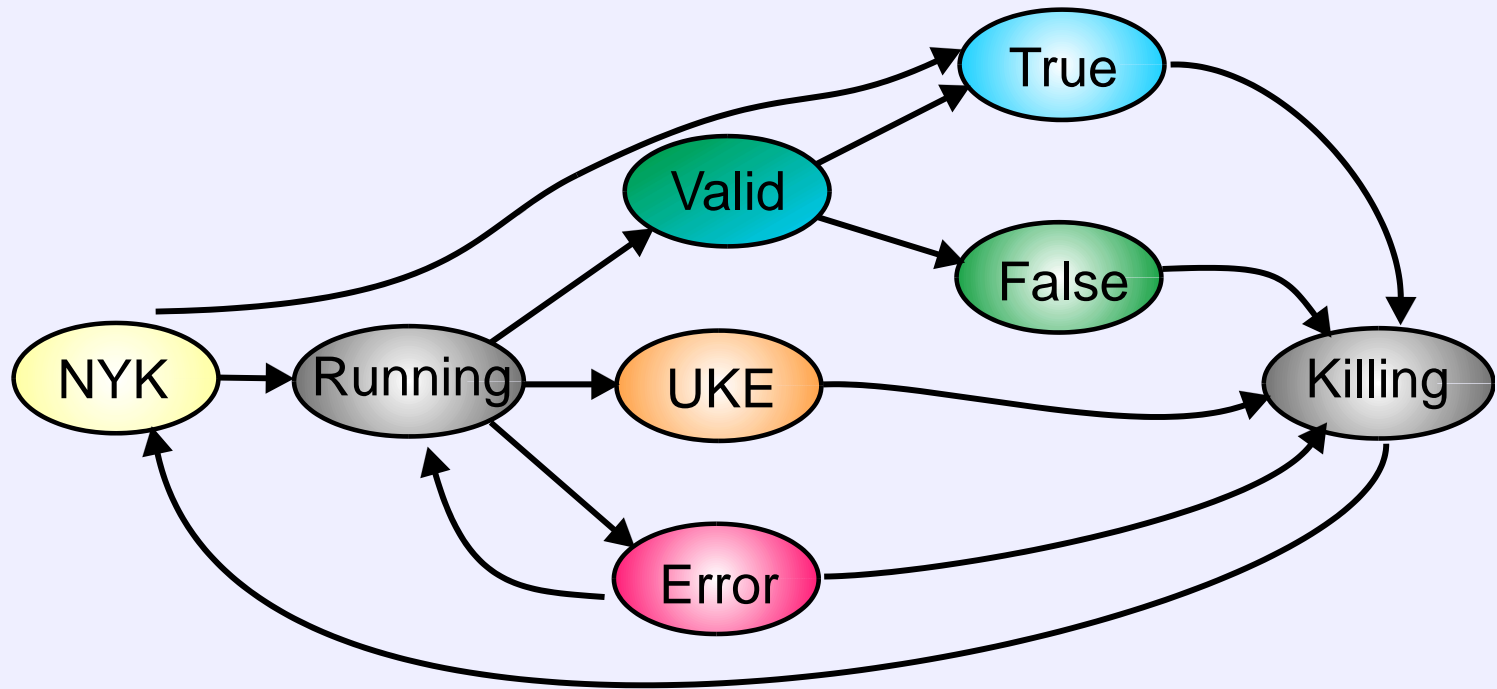


# A Flexible Finite State Machine

$$A + B = C$$



# Signalling of States



**The Phases Of Five State Logic**



# What's Wrong with Sequential Machines?

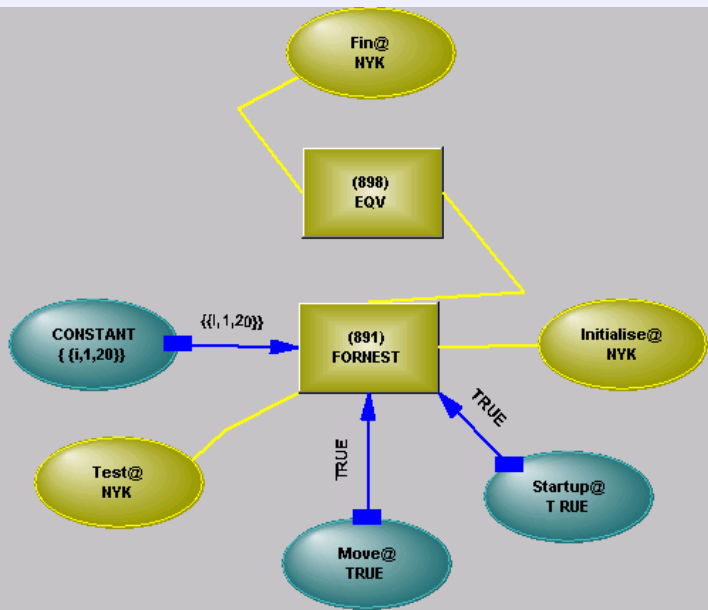
```
.....  
IF NOT ValidState  
    GOTO Out  
FOR i := 1 TO 10  
    DO BEGIN  
    .....  
    .....  
    END;  
Out: .....
```

```
-----  
Assign 1 -> I  
Jump1:-----  
-----  
-----  
JMP I > 10  
INC I  
JMP Jump1  
Out: -----  
-----
```

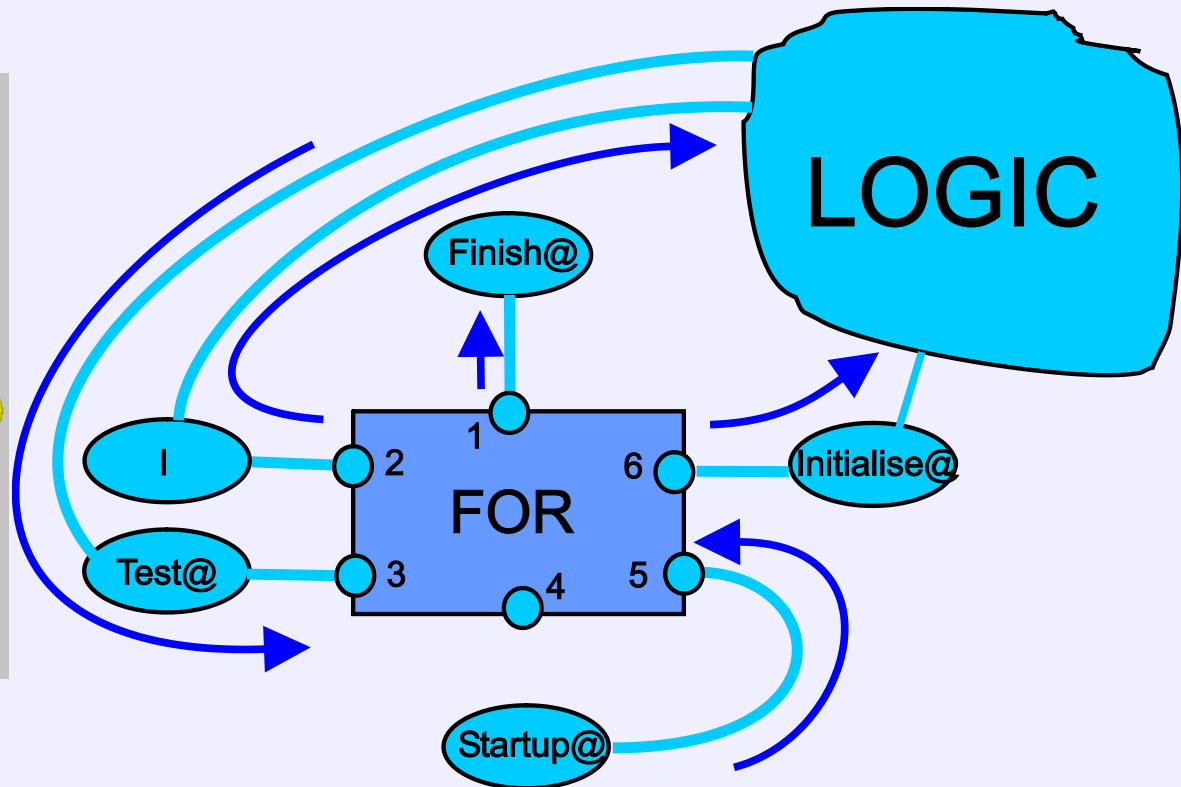
We would like to run the FOR loop only when the machine is in a valid state, but we can't suspend our one processor, and we can't loop



# The Equivalent Finite State Machine



Animation



Startup will only occur when we are in a True state - we will bypass if in False state

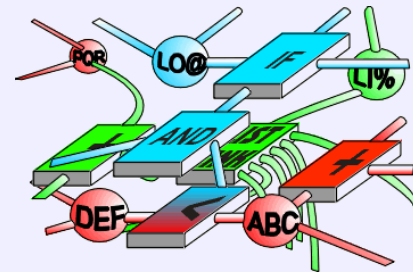


# What's the Point?

**We need to build machinery to handle every state transition - we can't use the crutch of sequenced instructions**

**If we do this, we can handle complex and dynamic phasing beyond the reach of programming**

**The structure can be modified on the run, it can see itself, adapt and extend itself, it can backtrack out of a scenario it constructed**



# We Also Need to Conserve Meaning

What does an EQUALS sign mean?

$$A = B + C$$

What about IF A = B THEN C = D

If we make IF...THEN... mean logical implication then the two EQUALS signs mean exactly the same thing

Why is this important - because we are not using sequence anymore - we are not programming, but building structure



# Knowledge Is Structure

## What does that mean?

Knowledge isn't rules that work from the outside, but structure that works from the inside, modelling the world and providing predictions of behaviour in previously unseen situations.

## What sort of structure does it imply?

The structure can't be directed to any purpose, and the structure needs to adapt itself to a changing environment - the structure needs to be active.





# How Is This Different?

The paradigm is based on:

- Connection
- Activity
- Visibility

It is an active constructionist approach, and from that flows extensibility, and the ability to manage complexity



# *Is It Really So Different*

We are asserting that knowledge can only be captured in active structure - structure that is capable of adapting itself to its environment.

Efforts at capturing knowledge in static structure founder on two reefs - the pieces of structure will not fit together statically, and an algorithm that could manage their combination would be more complex than the combination of the pieces, and is thus both unmanageable and unreachable from text.

Active structure avoids both problems - the pieces adapt to each other, and the behaviour of the combination is managed by the interaction of the pieces.

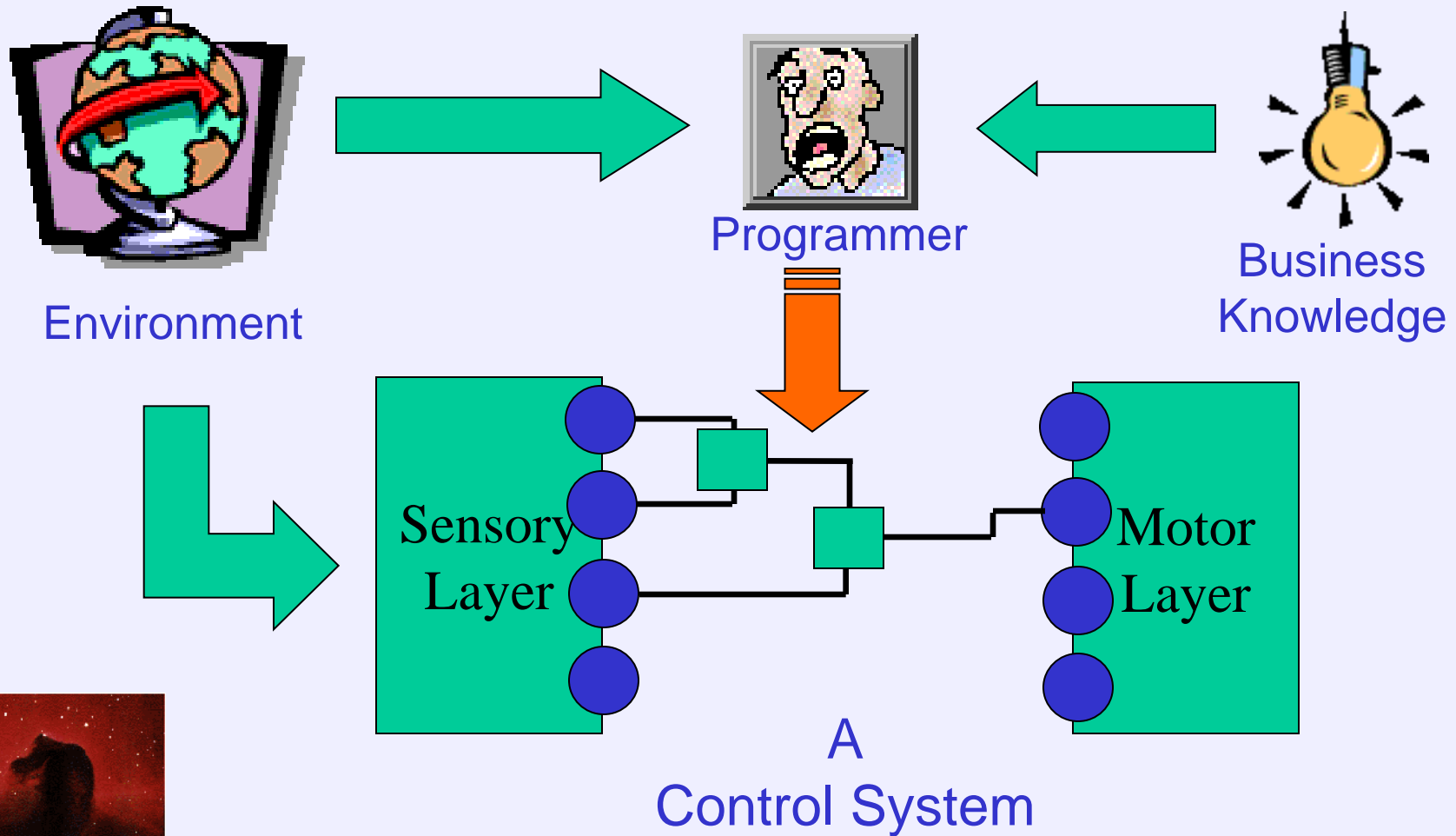


# What's Wrong With Static Structure

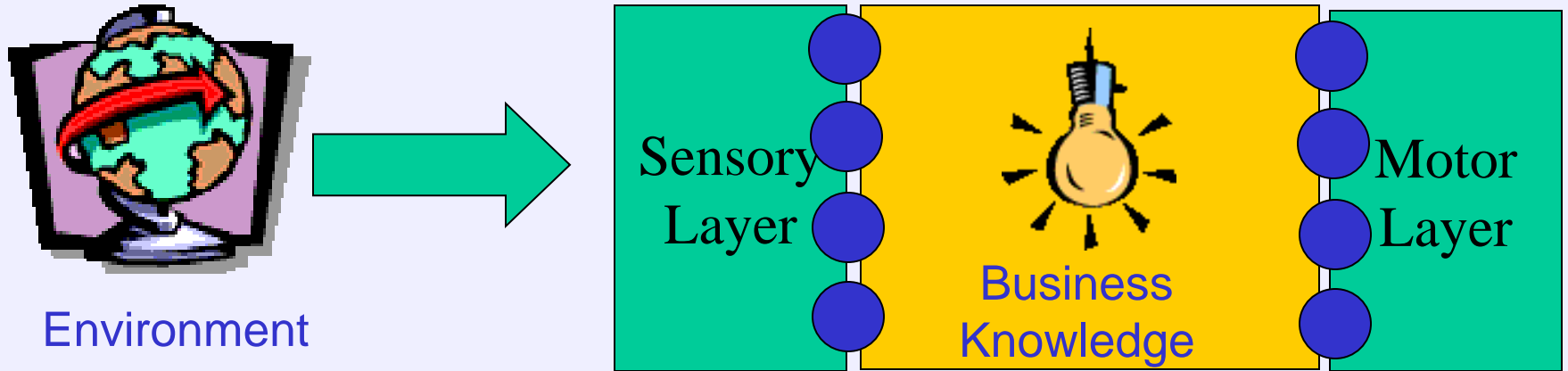
- The inferential structure that gave it birth has drained away
- Usually rooted in a hierarchy
- Neither self-sustaining, nor adaptive with other structures
- Requires an external active element which “understands” it



# Inferential Structure Drained Away



# When Inferential Structure Remains



Business knowledge is used as a cognitive layer, active and capable of rearranging its structure to adapt to changing circumstances, including changes to itself.



# What's Needed

Knowledge Structure  
Dynamic Construction & Destruction  
Constraint Reasoning  
Structural Backtrack  
Machine Learning



# Combining Structures

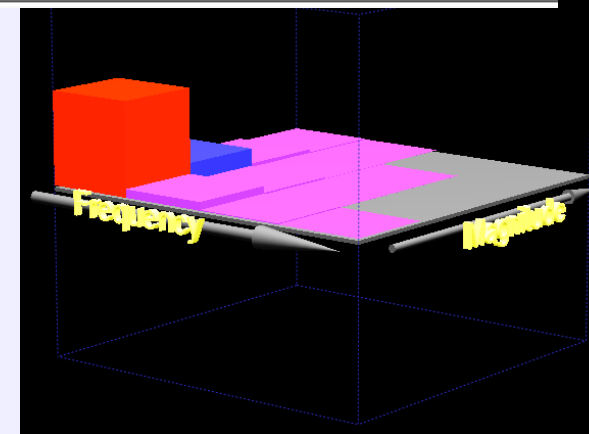
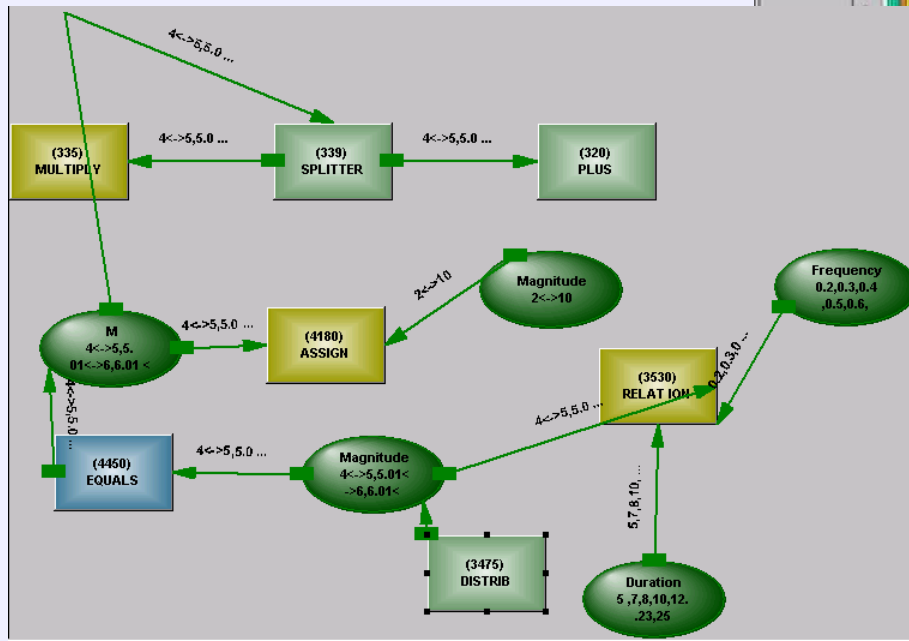
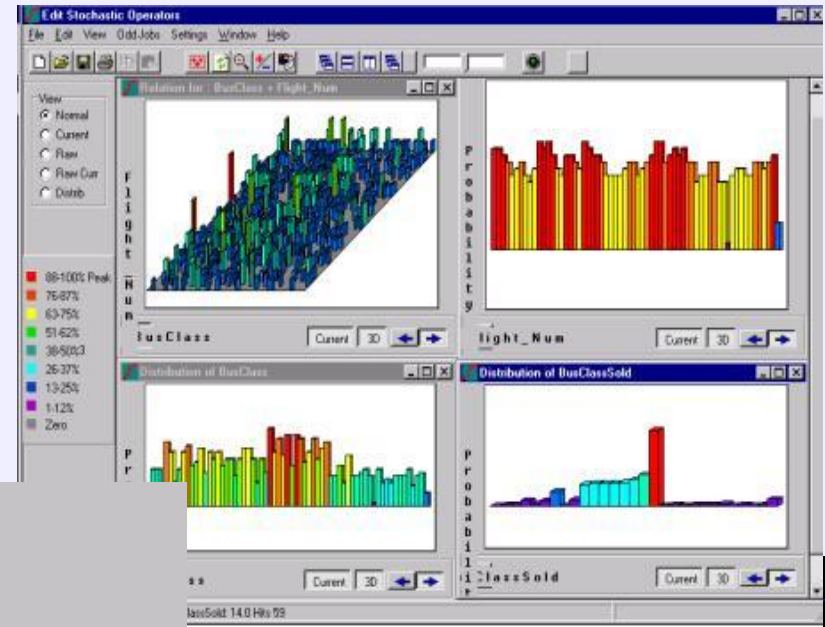
It may seem simple, but the ability to combine knowledge structures is a very strong constraint on how the knowledge is structured.

Each individual structure can't have a beginning or an end, and can't have an external algorithm which is needed to make sense of it - each structure has to be active, carrying all its own operation and phasing. Only then can the pieces build into larger assemblies which also carry the same properties.



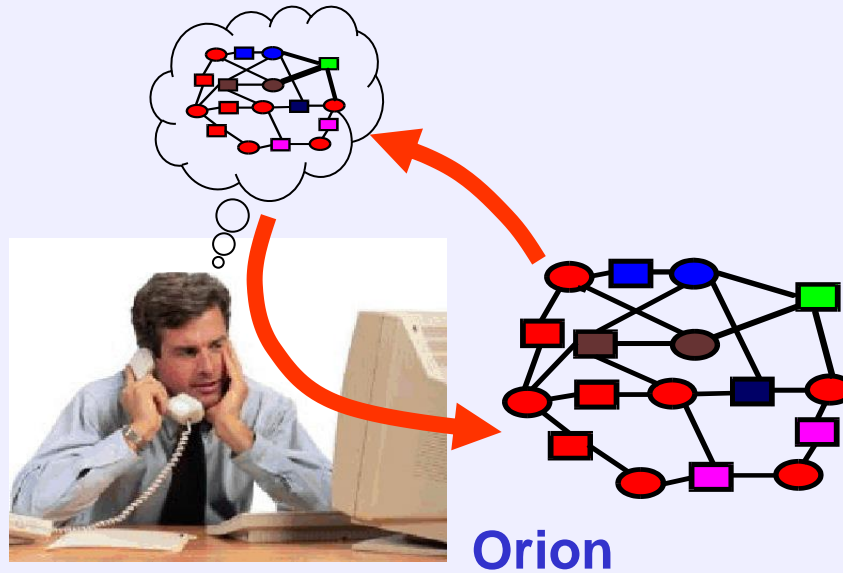
# Simple Machine Learning

Simulate scenarios, remember successful approaches in probabilistic relations inside the machine





# The System as a Thinking Tool



Using active structure to represent knowledge, the user can turn what they know into a shareable and reusable piece of knowledge by describing relations among objects - the structure can use the relations any way it needs to, when it needs to.

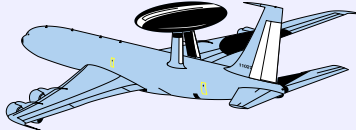
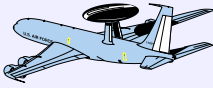
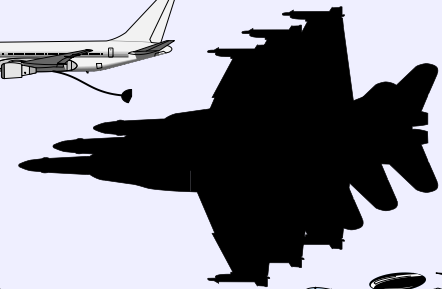
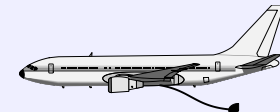
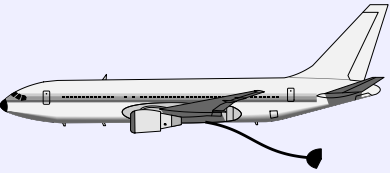
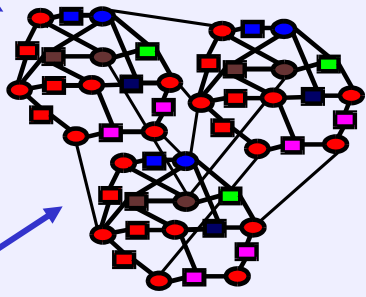
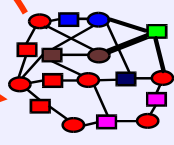
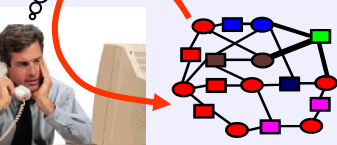
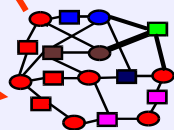
Users can refine their thinking by observing a model of it in operation.



# Combining Knowledge Domains

Strategy

Models from different domains can combine themselves automatically - based on the presence of context and control structures within the contributions.



# *The Rise of the Machines*

Machines changing their internal topology in response to the environment - that doesn't sound good.

Machines do it all the time. A plane lowering its undercarriage is a machine changing its topology.

We are talking about soft machines here, with internal resources to build structure, but a PLUS changing its connections doesn't sound too threatening.

