Temporal Recognition

Version 1.0
20th March 2018

Adaptron Inc.

Temporal recognition is the perception of time based patterns.

Content

- Temporal pattern recognition
- Spiking and counting
- Binons states
- Temporal activation tree
  - Short Term Memory
- Polling versus event driven

This presentation describes how a spiking version of binons interact to perform temporal pattern recognition.

The process involves binons counting spikes from source binons, changing state and firing.

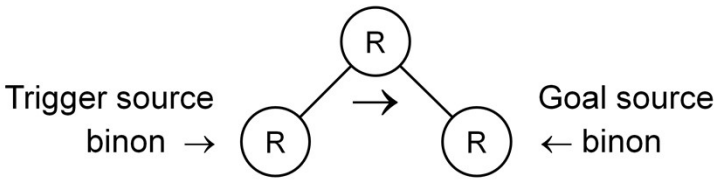
Then the binon firing stimulates all its target binons.

Time Based Recognition

- Sequential habits
  - Speech recognition
  - Music recognition
- Requires a short term memory
  - To form temporal patterns
  - To detect repetition
  - Long duration patterns persist for longer time

Temporal Patterns

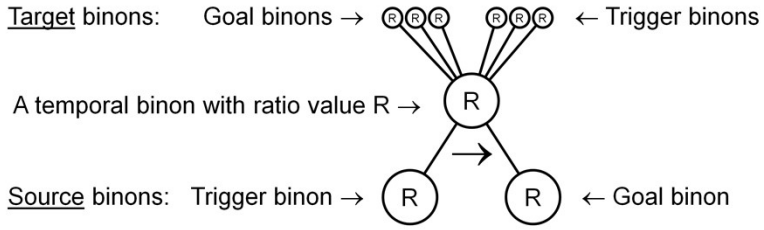
- Represented by temporal binons
- Shown using the arrow sign ($\rightarrow$)



The arrow for temporal recognition.

A Temporal Binon

- Trigger source binon activates the binon
- Goal source binon fires the binon



- When fired
  - It stimulates all its target binons

Source binons are closer to sensors and thus more general.

Source binons are combined to form a target binon.

This is a compositional structure, binons are added together.

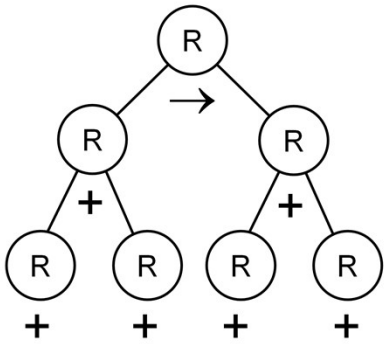
Using two source binons results in a binary hierarchy.

But multiple target binons produces a lattice network.

Trigger source binons are on the left and Goal source binons are on the right.

Temporal Patterns

- Source binons may be spatial patterns
  - Multiple sensors



Adaptron Inc.

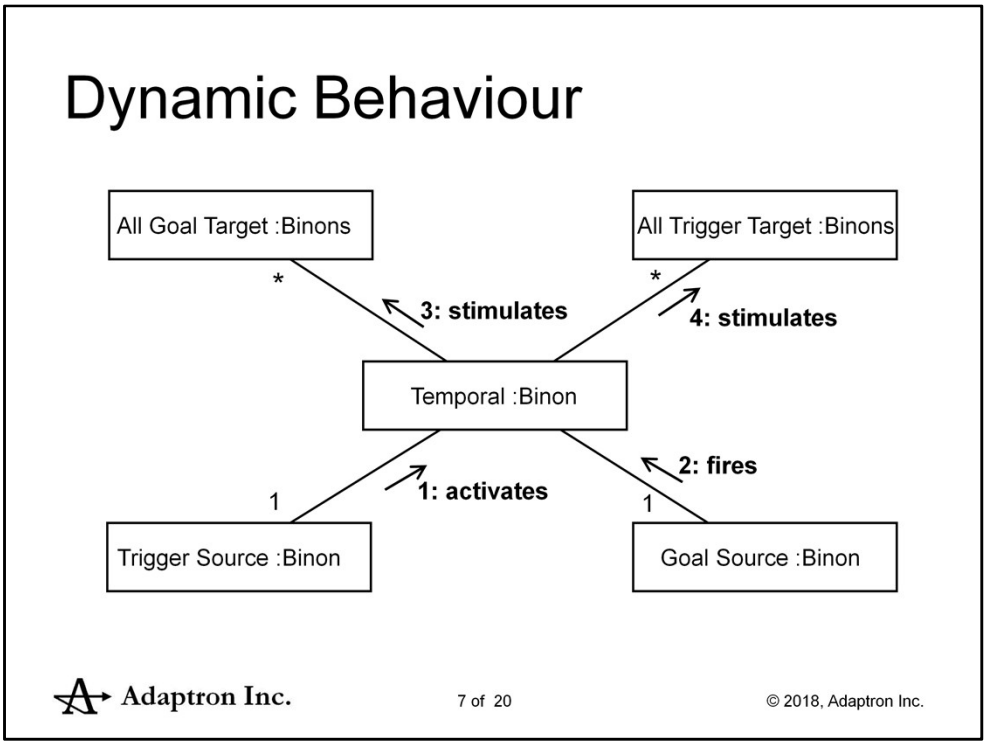
6 of 20

Temporal patterns are aggregations of spatial patterns if there is more than one sense and/or sensors involved.

Spatial patterns are only composed of other spatial patterns.

You can not have a spatial pattern where any of the sub patterns are temporal.

Or in simpler terms things get combined spatially first and then temporally.



This is a UML Communication diagram.

This is the same dynamic behaviour for spatial recognition.

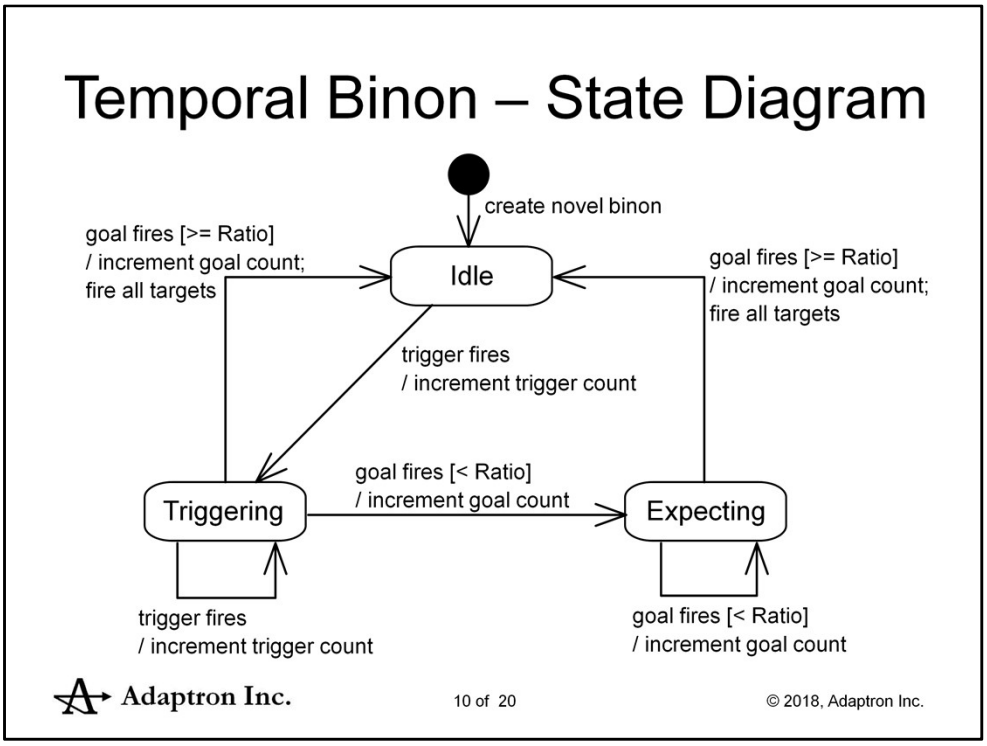
Except the two source binons fire in sequence rather than in parallel.

Spiking Architecture[1]

- Each binon counts its stimulations from trigger and goal source binons
- Compares the ratio (trigger / goal firings) with its expected ratio
- Fires if it matches – pattern recognized
- Stimulates all its trigger and goal target binons

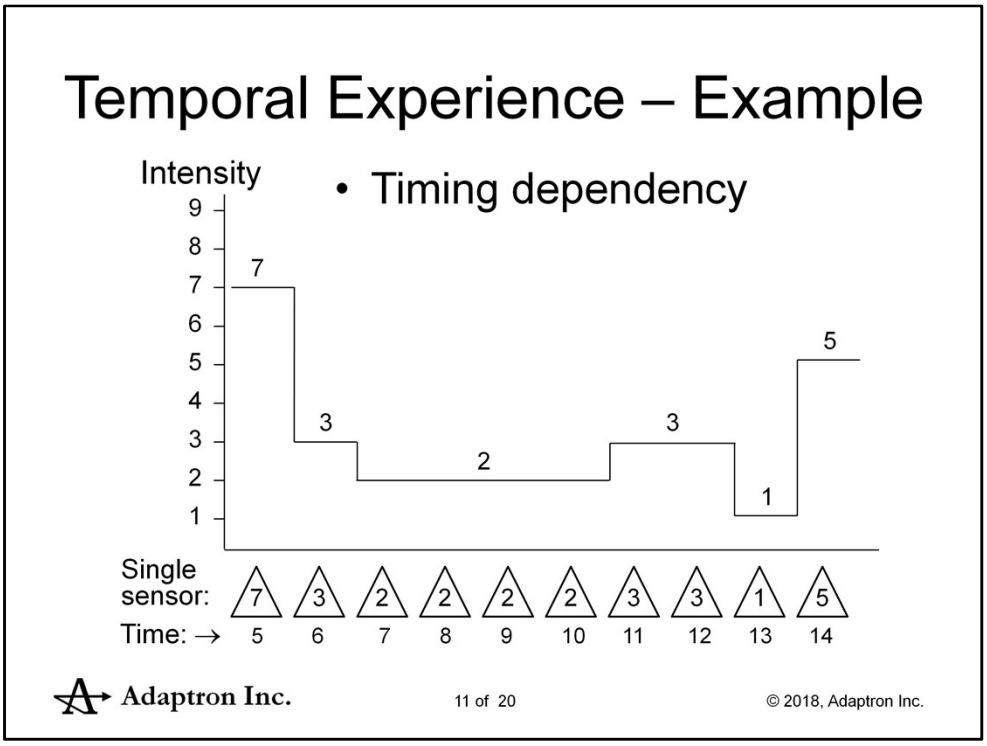
Binon States

- Idle
  - Waiting for the trigger source to fire
  - It does not fire upon binon creation
- Triggering – trigger source is firing
  - Waiting for the goal source to fire
- Expecting – goal source is firing
  - Waiting for its ratio to be reached



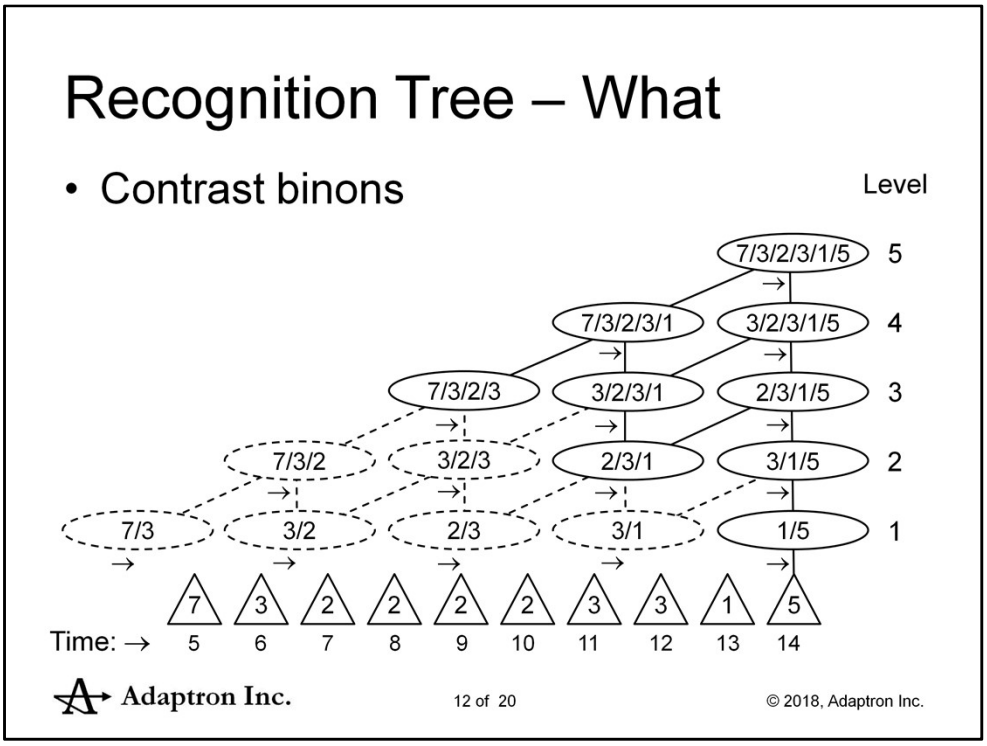
This is a UML State diagram. Also called a state transition diagram.

I have implemented this in software and have it successfully recognizing Morse code.



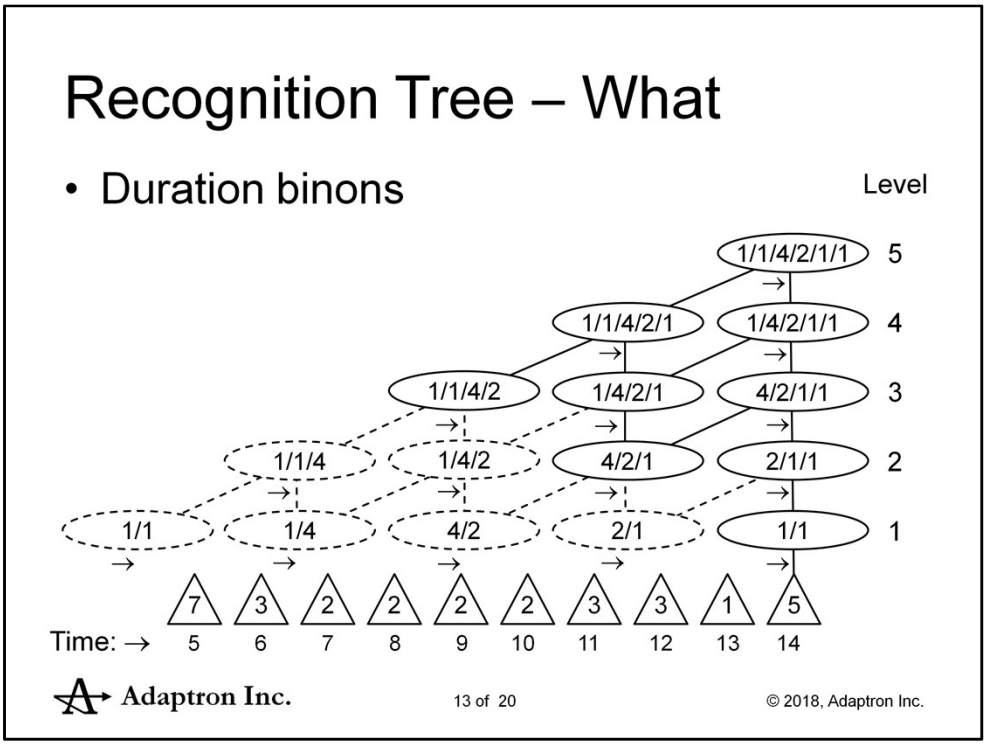
The time shape of the pattern is what has to be recognized.

The time shape is a combination of the relative intensity (contrast) and durations patterns.



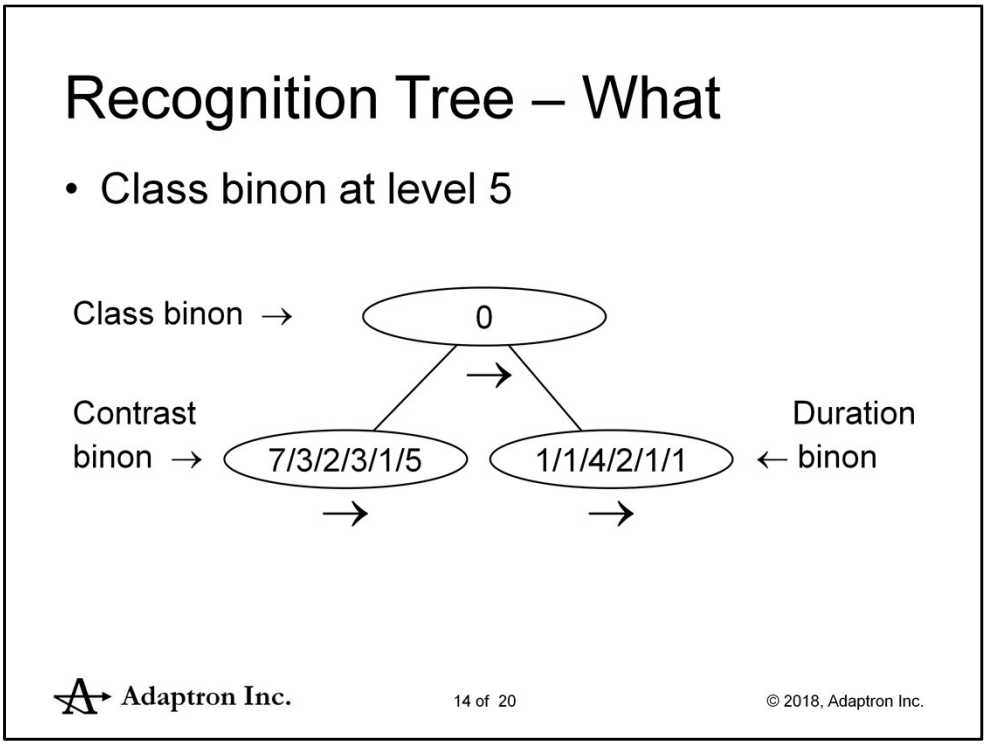
It needs to keep longer duration patterns (at higher levels of complexity) for longer periods of time to detect if they have repeated.

[Level 1 binons contain values representing the ratios, higher levels of overlapping binons actually contain zero values representing a 1/1 ratio. The multi-valued ratios on the diagram are for displaying what the binons represent based on their combination of source binons.]



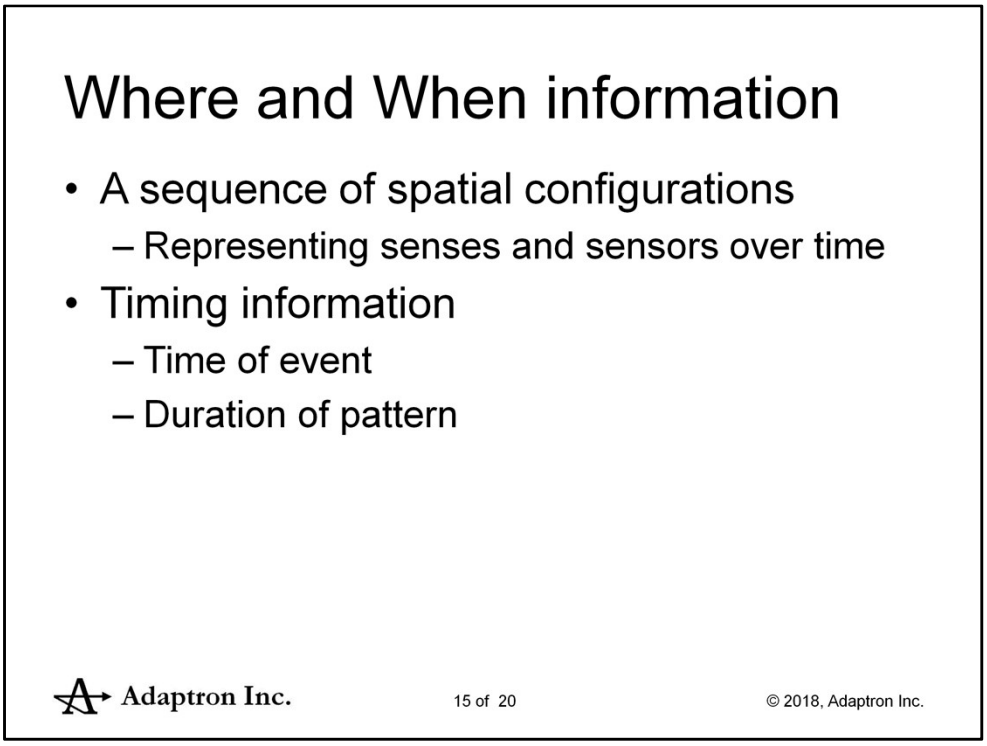
It needs to keep longer duration patterns (at higher levels of complexity) for longer periods of time to detect if they have repeated.

[Level 1 binons contain values representing the ratios, higher levels of overlapping binons actually contain zero values representing a 1/1 ratio. The multi-valued ratios on the diagram are for displaying what the binons represent based on their combination of source binons.]



Combine the Contrast and Duration (Time shape) binons to get the Class binon. This is a simplified version; the complete tree also needs the repeat quantity pattern (binon) and the separation pattern (binon) at every level.

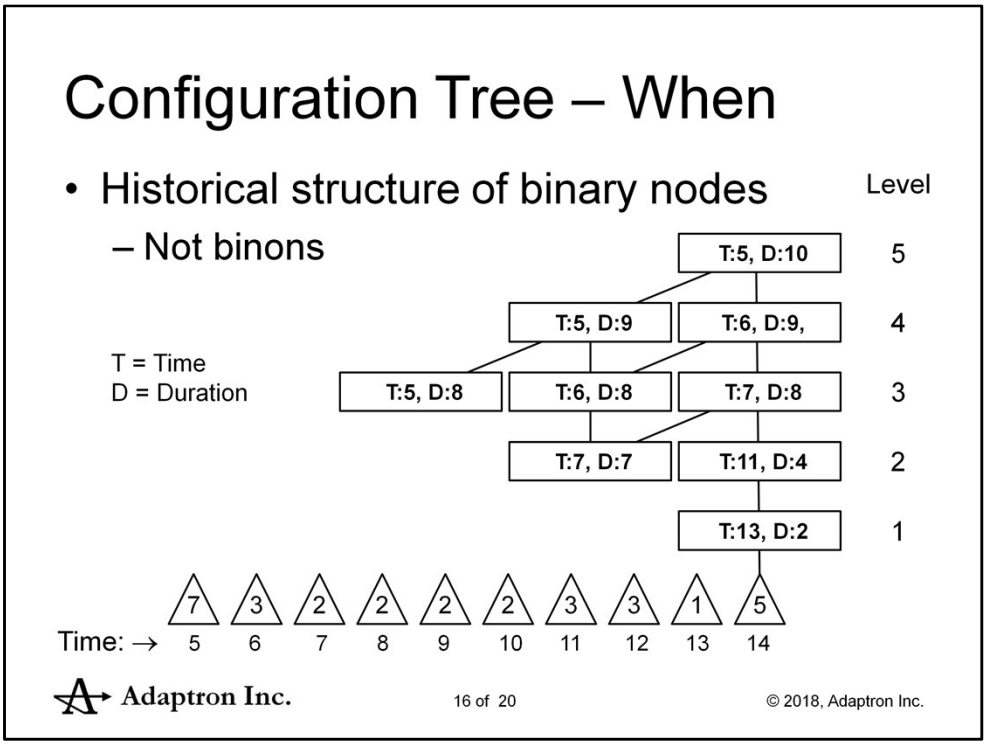
[The two source binons actually contain zero values representing a 1/1 ratio. The multi-valued ratios on the diagram are for displaying what the binons represent based on their combination of source binons.]



The configuration information is a sequence of the spatial configurations with timing information.

The timing information is a time stamp and a duration.

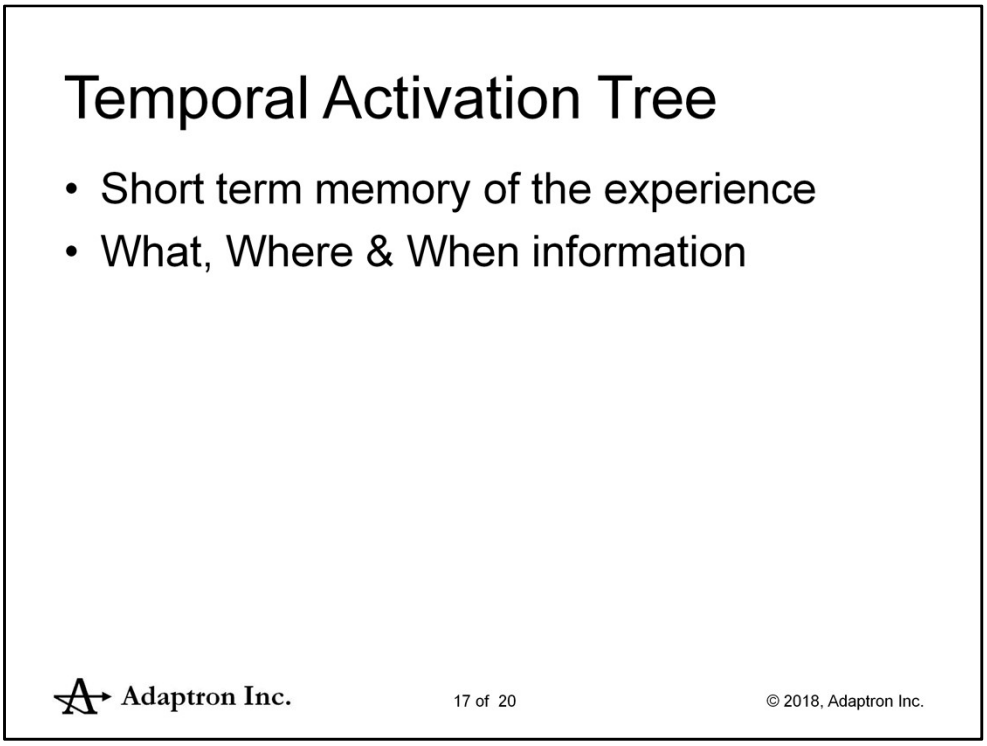
Check out the Experiences presentation for more detail about the spatial configurations of senses and sensors.



This is a simplified version; the complete tree also needs the sense and sensor identification for each entry.

That is the spatial configuration information.

The Morse code recognition that I have been doing only involves one sense and one sensor so that information has been left out.

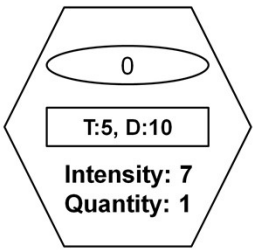


The temporal activation tree brings it all together. This is the Short Term Memory.

Temporal Activation Tree

- Same historical binary tree structure
  - Not binons
- Each node contains:
  - + Class binon
  - + Spatial & Timing configuration
  - + Activation readings

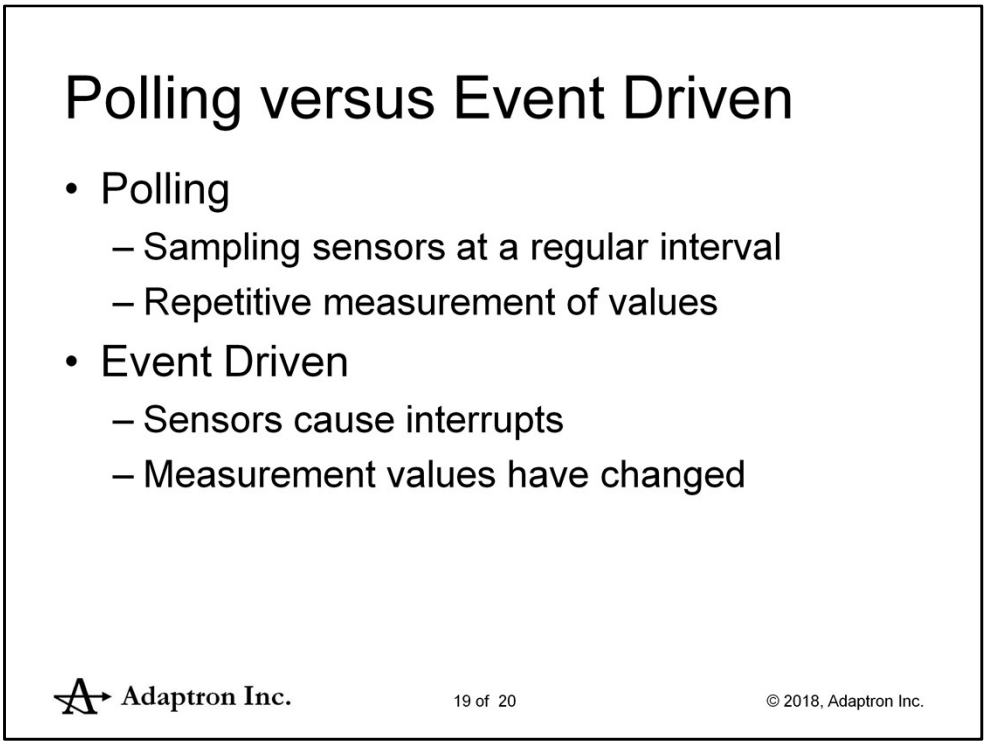
Level 5 example



Each activation tree entry contains a reference to the class binon recognized at that point in the experience.

The senses, sensors and timing configuration information is also referenced.

And the activation tree keeps the absolute values of intensities and repeat quantities.



There are two approaches to obtaining sensor readings over time; polling and event driven.

If sensors are being continuously polled for their readings then the previous and current measurements must be compared to detect a change in value.

The duration between changes can be calculated. When no change is detected no recognition takes place.

In the event driven approach the sensors need to detect when there is a JND.

Then they cause an interrupt and provide the values as measured at the time of the event and the duration is calculated from the current event time minus the previous event time.

Both approaches work. The event driven approach is better if a single processor is performing conscious recognition, action and thinking.

Thinking can be interrupted by an external event.

References

[1] Qiang Yu, Huajin Tang, Kay Chen Tan and Haizhou Li, Rapid Feedforward Computation by Temporal Encoding and Learning with Spiking Neurons, IEEE Transactions on Neural Networks and Learning Systems, Vol. 24, No. 10, Oct. 2013