Unless Node

The unless node (Fig. 1) is an example of a complex node. The unless node will transmit an event arriving at the input port (the exciting event) to the output port, provided that no event arrives at the inhibitor input port (the inhibiting event) which satisfies the criteria specified by the window parameter and the condition parameter. These events can arrive at different times; either order is supported. The transit delays of arriving events must be within the window parameter limits to be accepted by the relevant port. If an accepted inhibiting event arrives and there is an accepted exciting event in memory, the condition parameter is evaluated. If an accepted inhibiting event arrives and there is no exciting event in memory, the inhibiting event will be held pending the arrival of an accepted exciting event, at which time the condition parameter will be evaluated. The condition parameter is a Boolean expression that can take both the exciting event and the inhibiting event as arguments. For example,

```
condition: input_event("device") = inhibitor_event("device")
```

will evaluate true if both events were generated by the same device (assuming this information is contained in the events). Any environment data (data store, fact store, and node attributes) can also be used in the condition expression. If the condition evaluates false, the exciting event is output via the output port. If the condition evaluates true, the exciting event is inhibited and is output via the inhibited output port if one is connected, or discarded if not. If the evaluation of the condition causes an error (e.g., if a referenced event attribute does not exist), the exciting event will be combined with the inhibiting event and output via the error output port as a composite event. If the error output port is not connected, the composite event will be logged. If the transit delay of the exciting event does not satisfy the window parameter transit delay window, the event is output via the fail output port. If the inhibiting event fails this test, it is silently discarded. The inhibiting event is never output except as a composite event as described above.

An event arriving at the reset input port causes any events held in the memory of the input (exciting) port to be output immediately via the fail output port, and any events held in the inhibitor input port memory to be silently discarded. The reset event is immediately output via the reset output port.
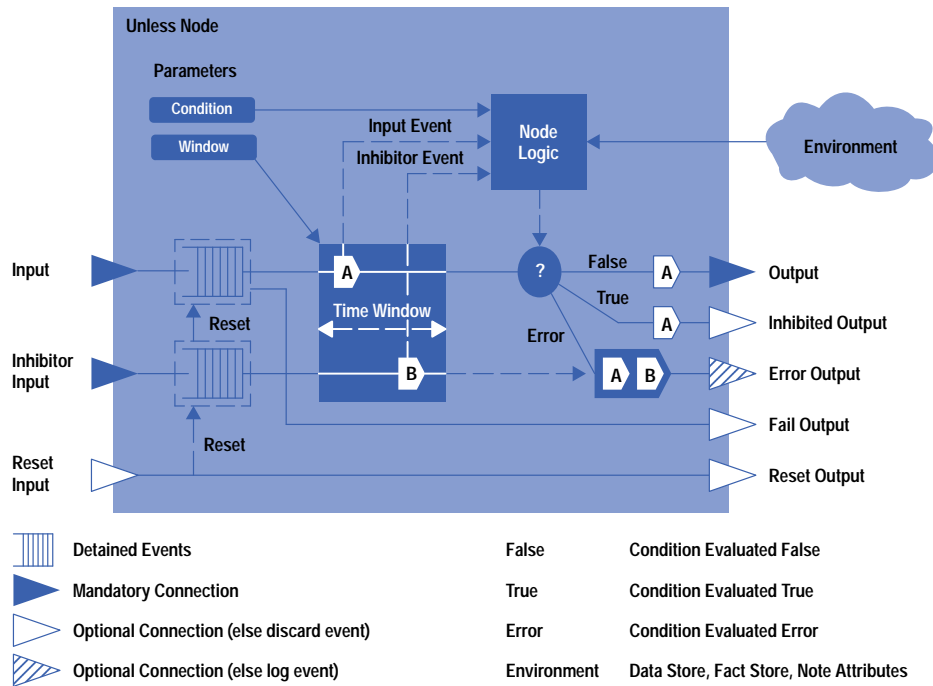

Fig. 1. *Unless node.*