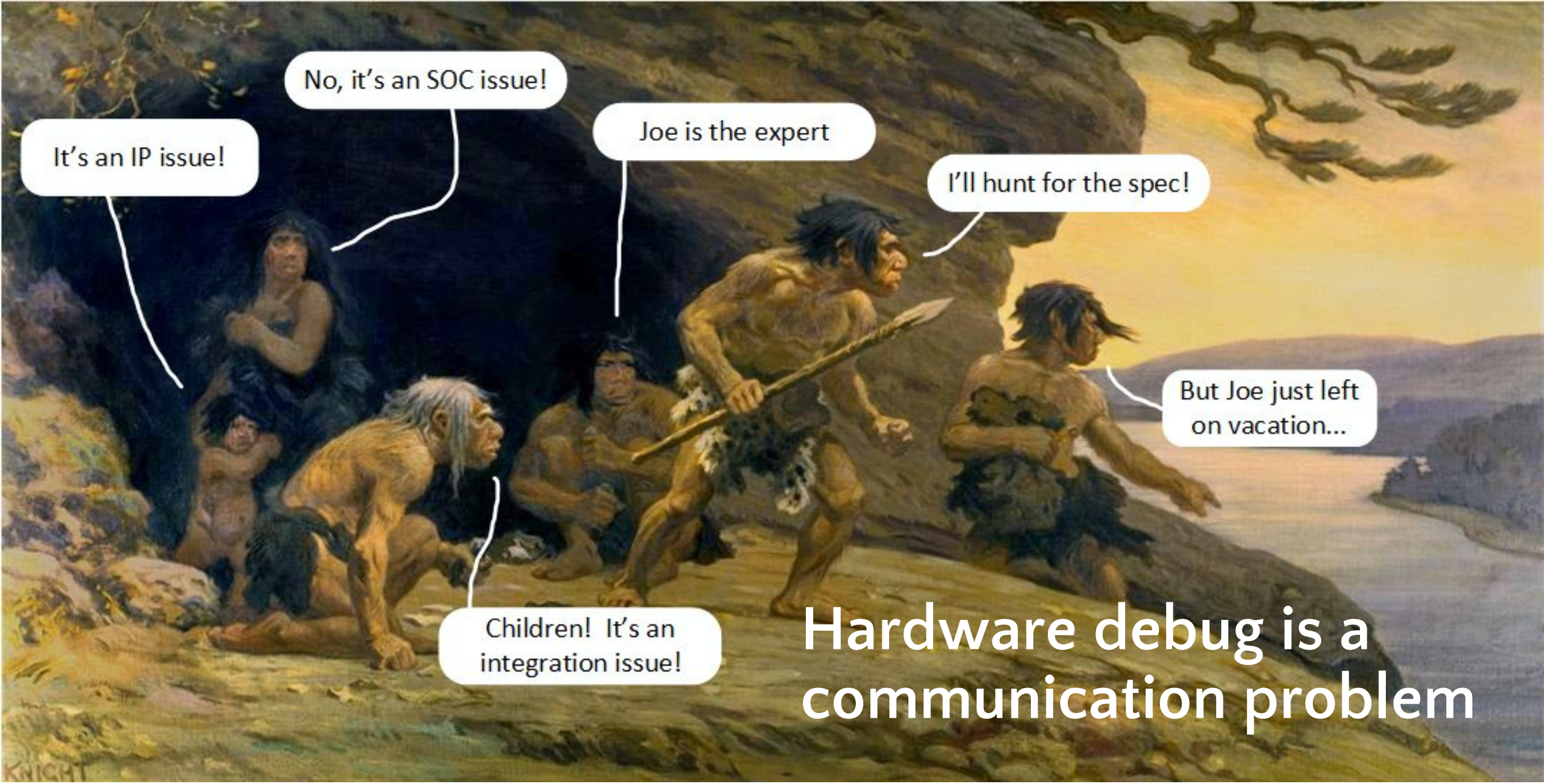
Debug Decision Trees

Erik Berg

Microsoft - Principal Engineer



Hardware debug is a communication problem

Successful Debug

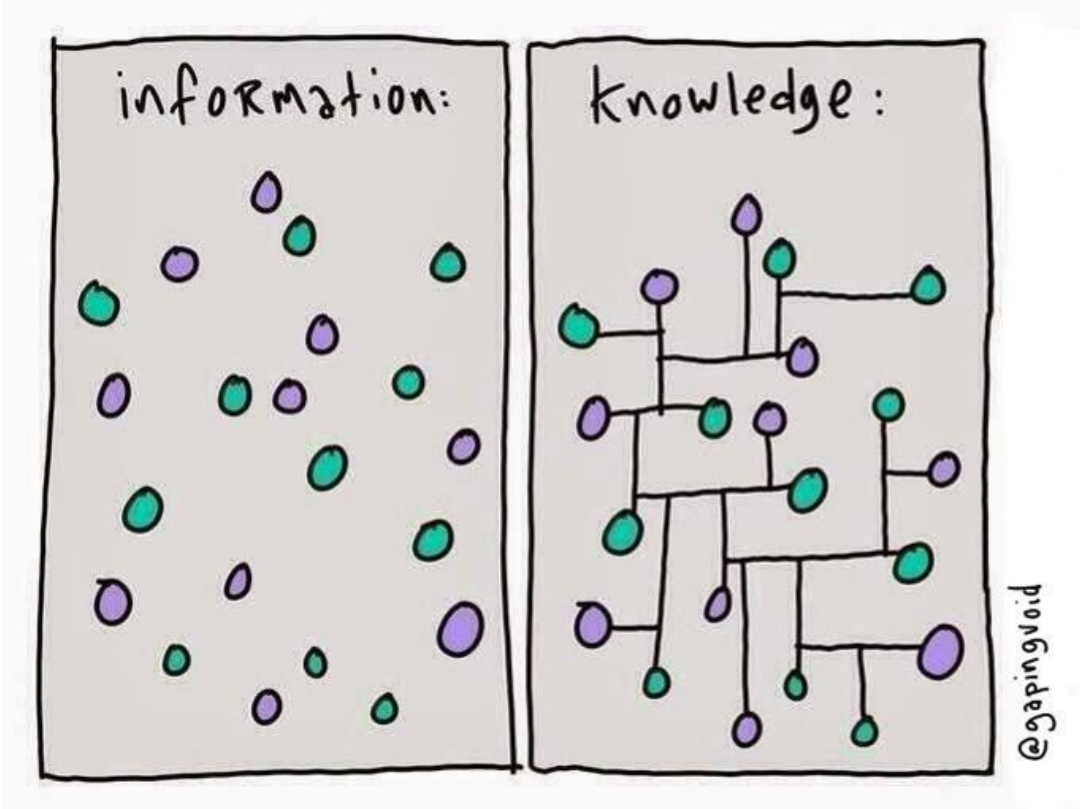
"The minimum sequence of observable events to deterministically define the path from the error message to the root cause."

Information: Observable events

- Testbench inform statements
- Waveform events
- Transaction log files

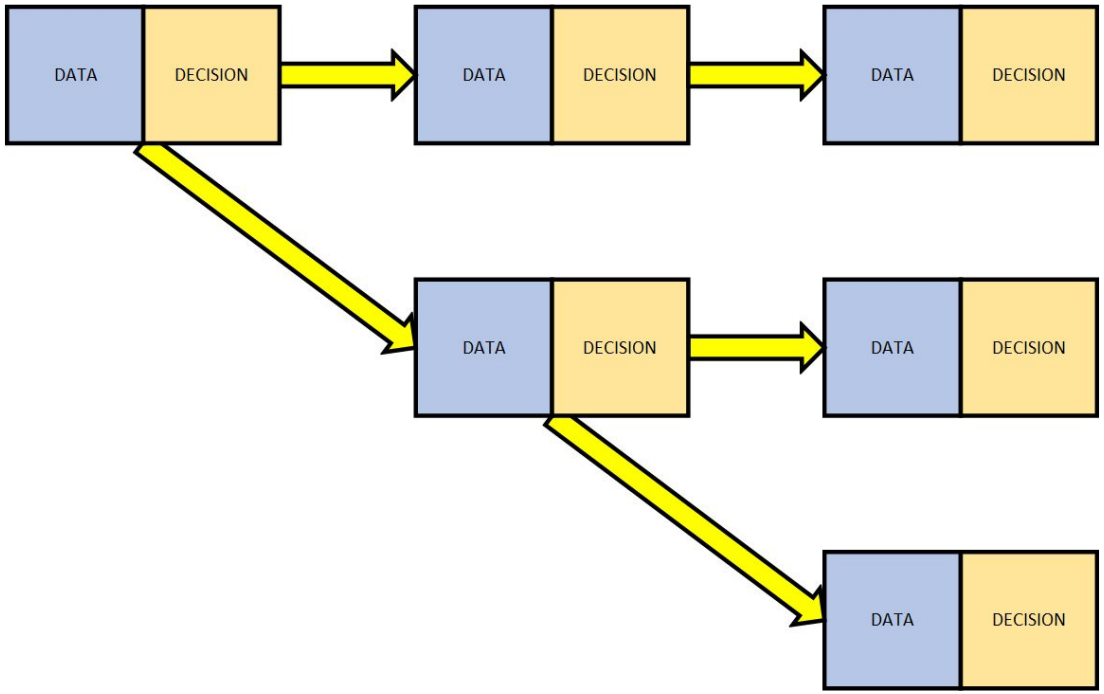
Knowledge: Connections between events

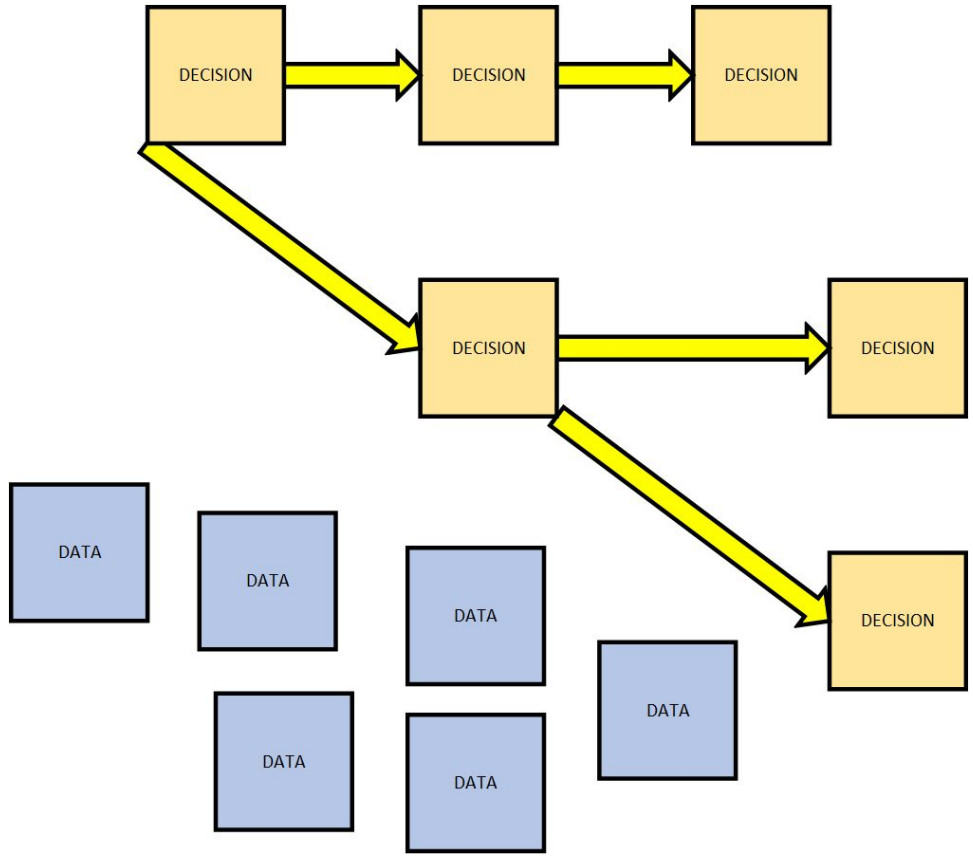
- Highly valuable expert knowledge
- Difficult to capture in text – context is key



ThenWhatTree

Separate data extraction from debug decision





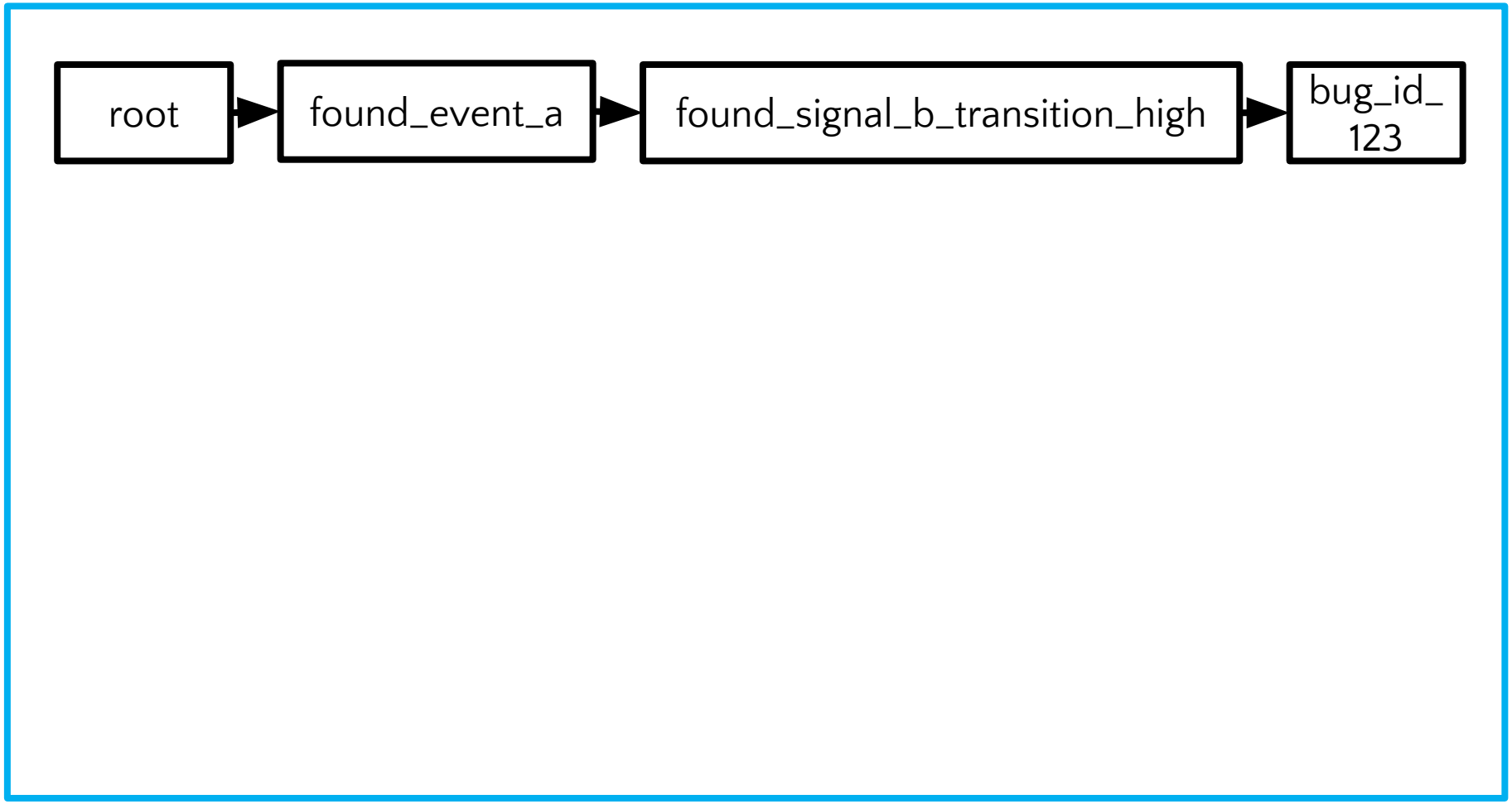
DDT Usage Model

Joe debugs a failure and can describe the root cause.

Creates the root node with required context

Adds the branch for the failure

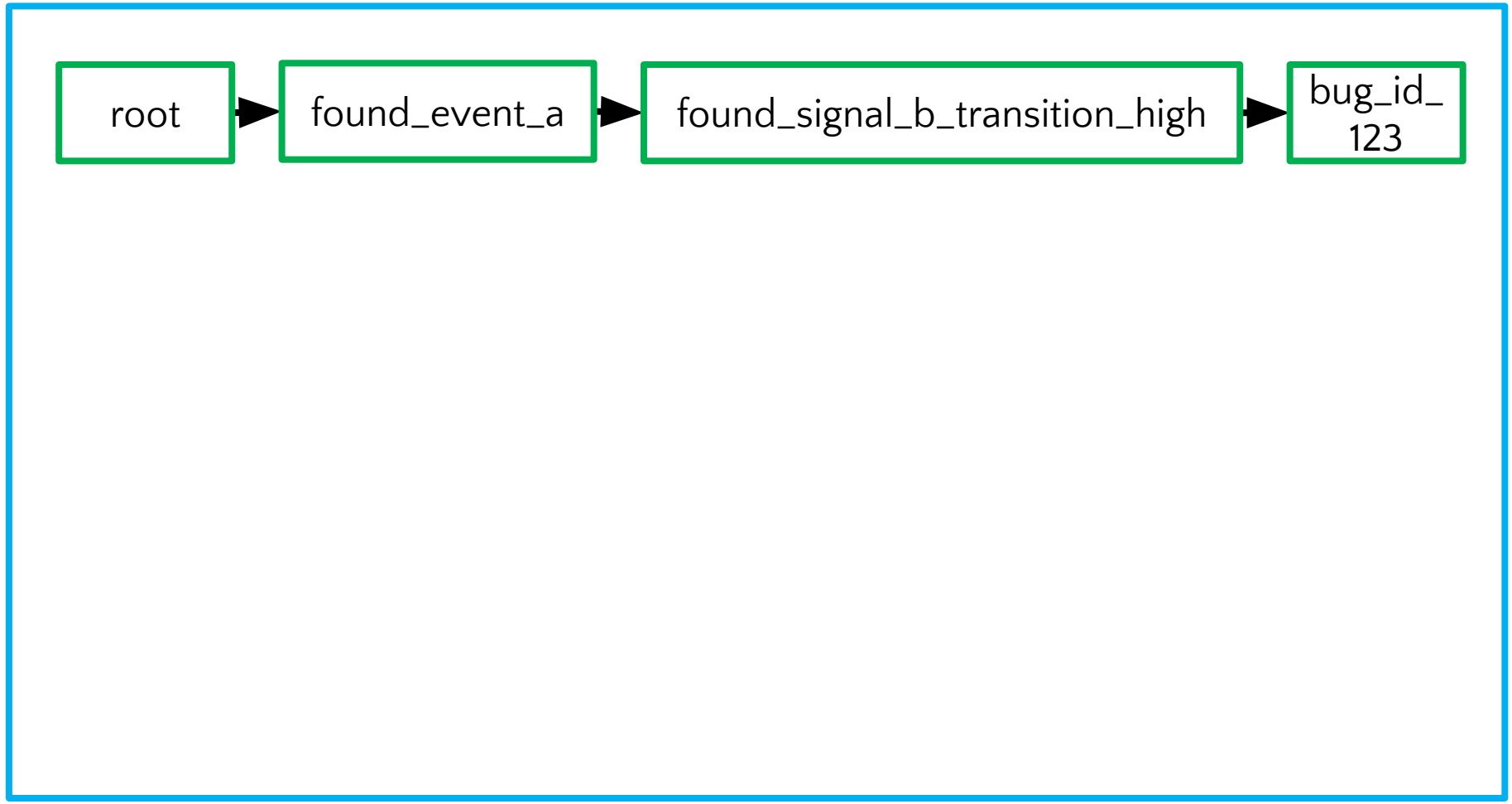
Generates .py executable



DDT Usage Model

The ddt .py executable is run as a post-processor against new failures

Match: no debug is required

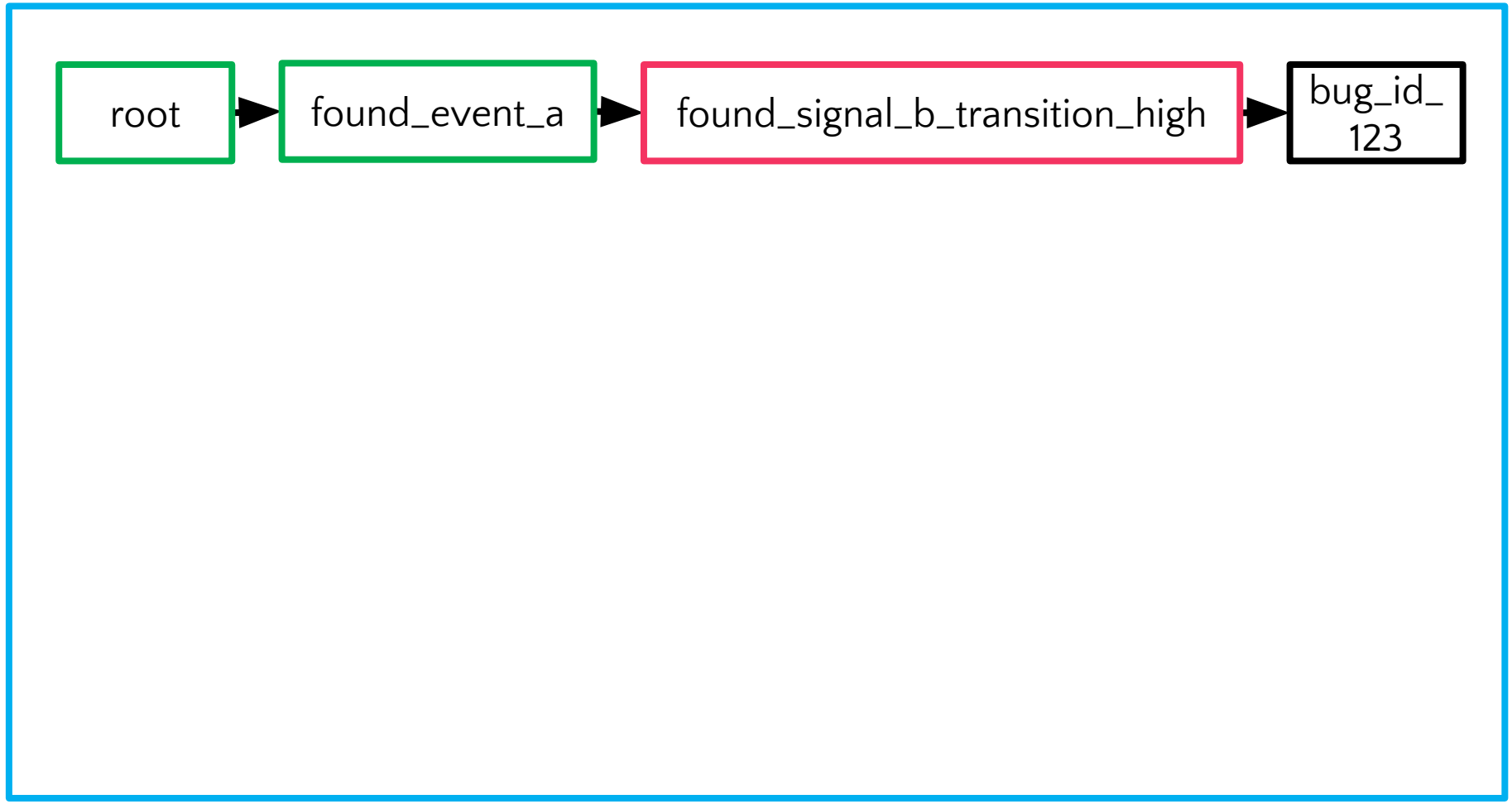


DDT Usage Model

The ddt .py executable is run as a post-processor against new failures

Match: no debug is required

No match: debug as usual



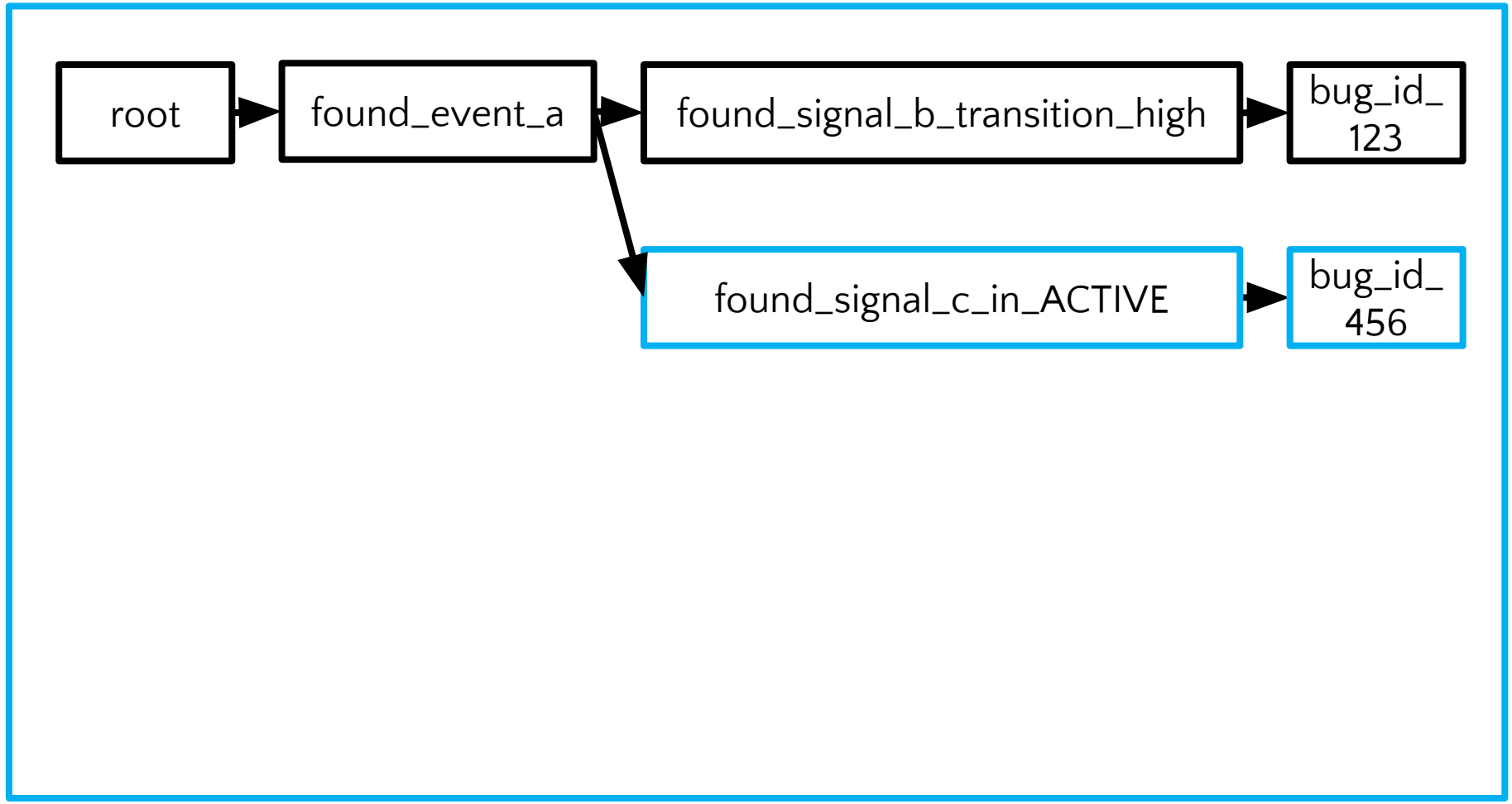
DDT Usage Model

Joe's coworker debugs a new failure

Extends the root node if new context is required

Adds the branch for the failure

Generates new .py executable

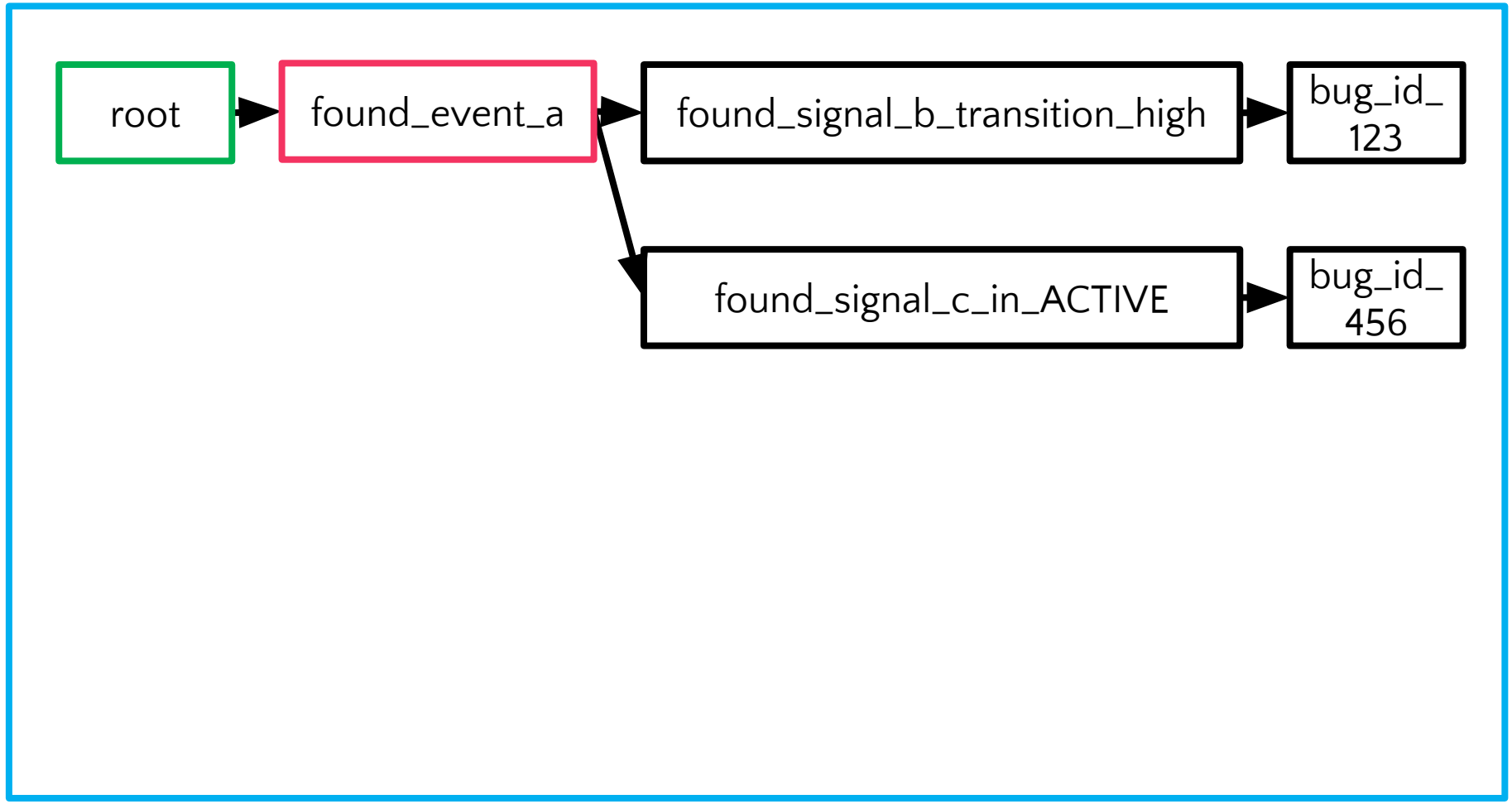


DDT Usage Model

The ddt .py executable is run as a post-processor against new failures

Match: no debug is required

No match: debug as usual



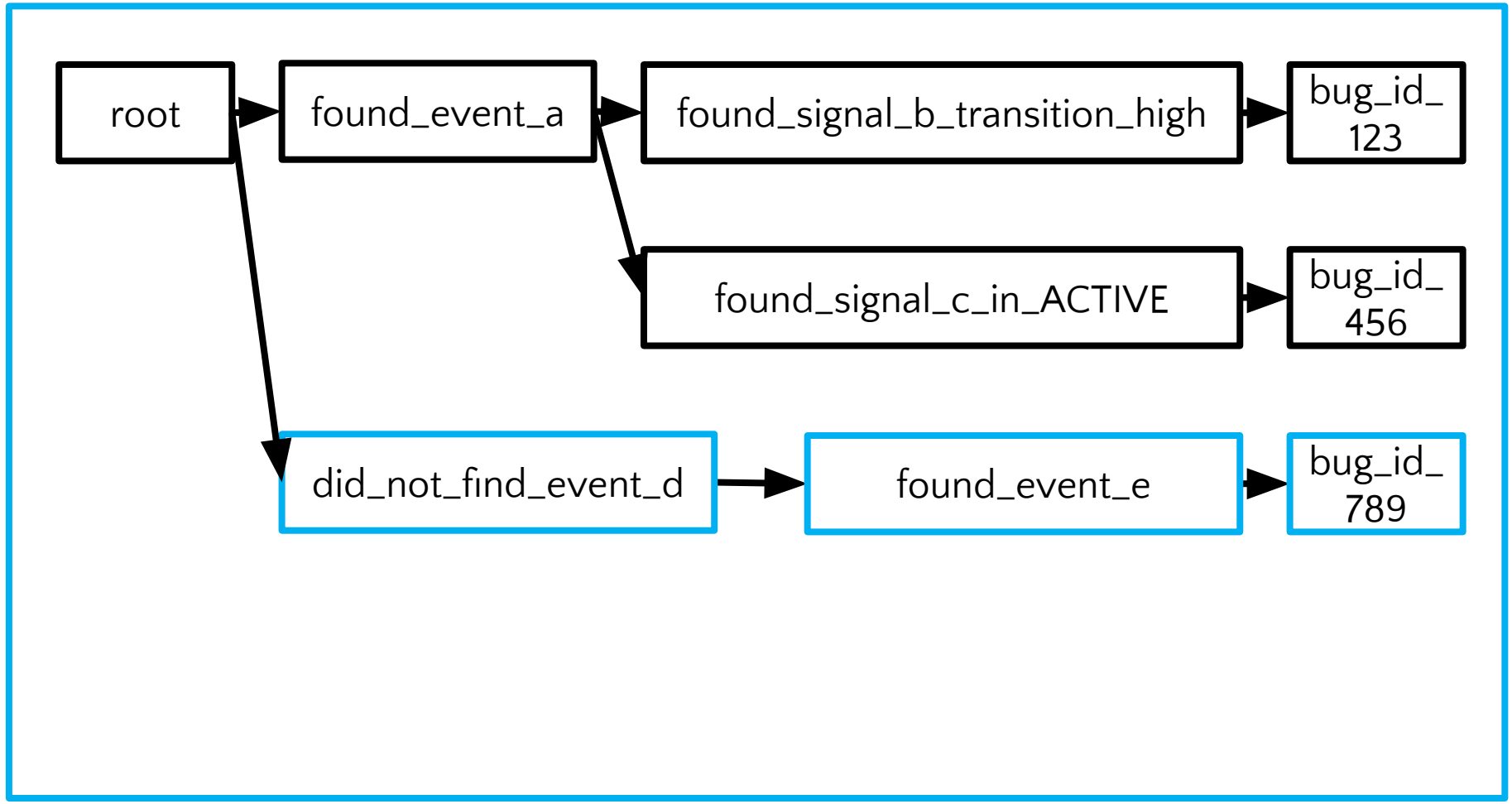
DDT Usage Model

Joe's coworker debugs a new failure

Extends the root node if new context is required

Adds the branch for the failure

Generates new .py executable

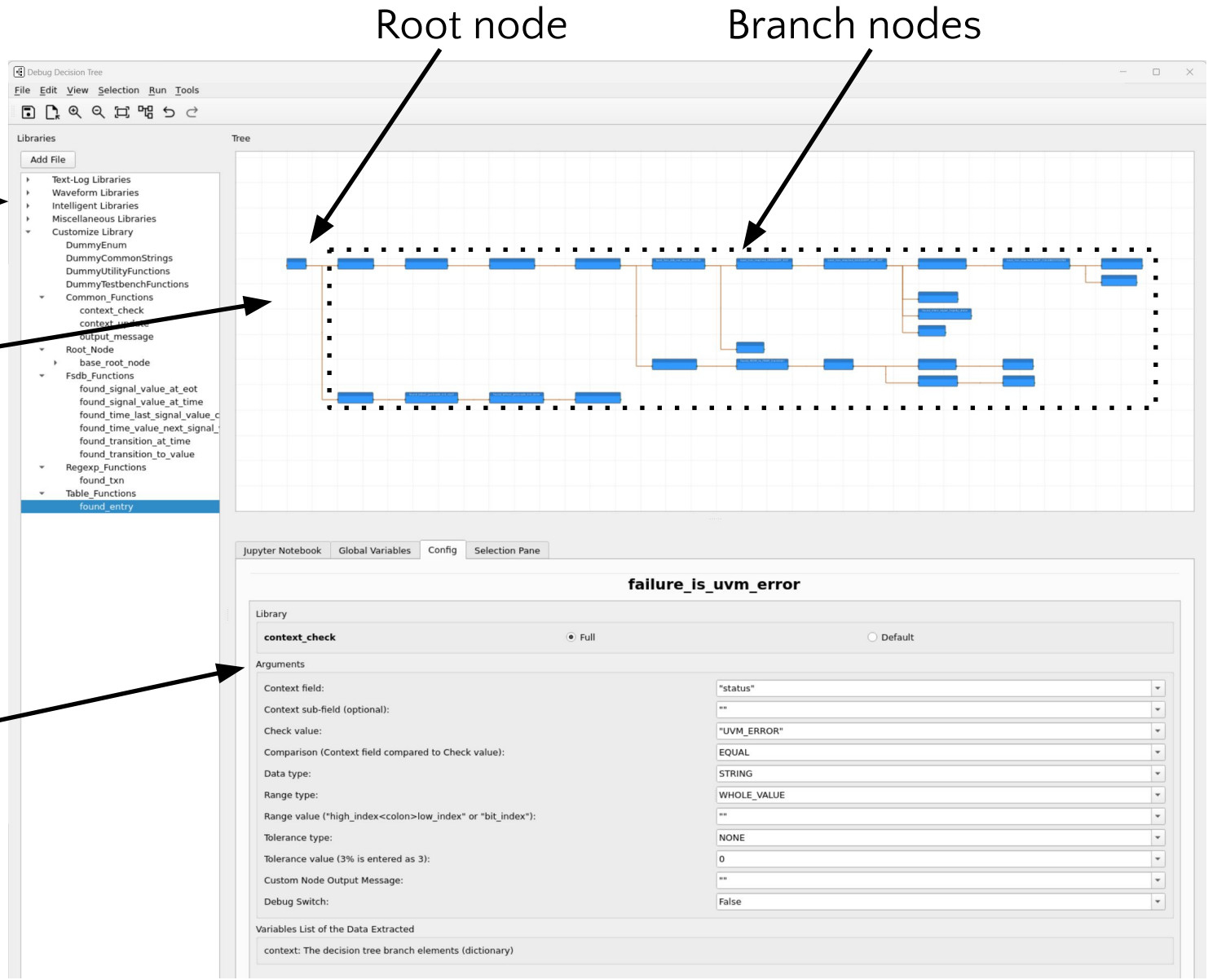


DDT GUI

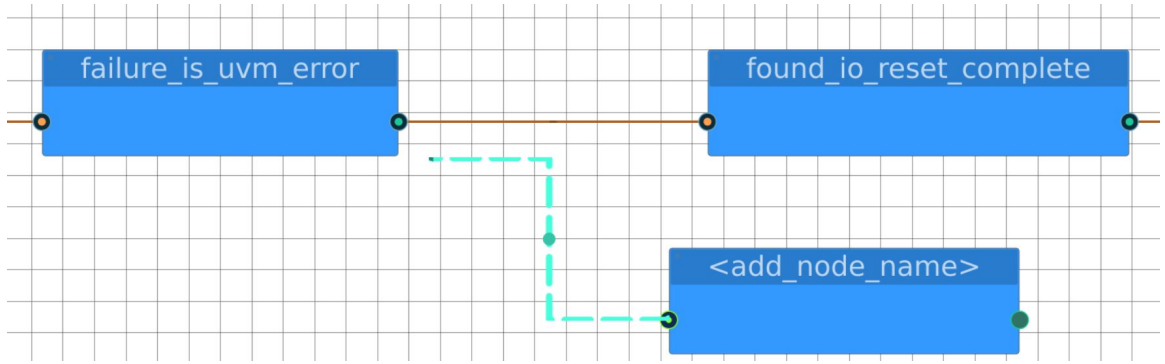
Node template libraries

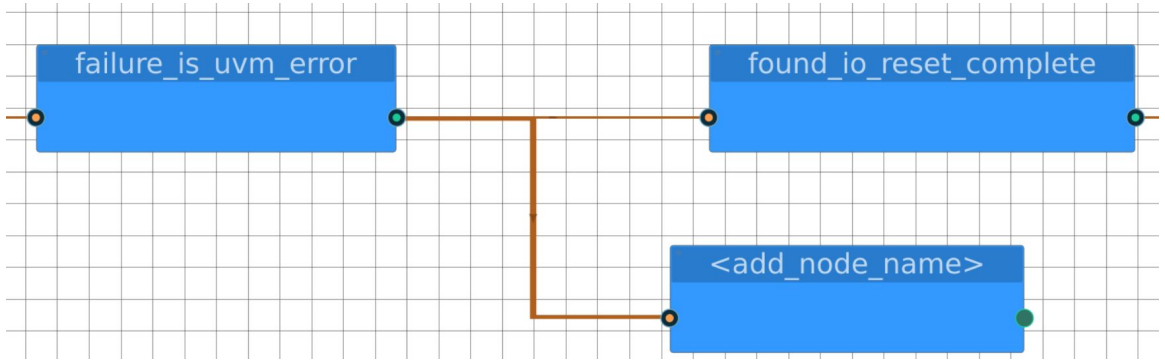
Decision tree structure

Node parameter inputs



Connecting New Branch Nodes



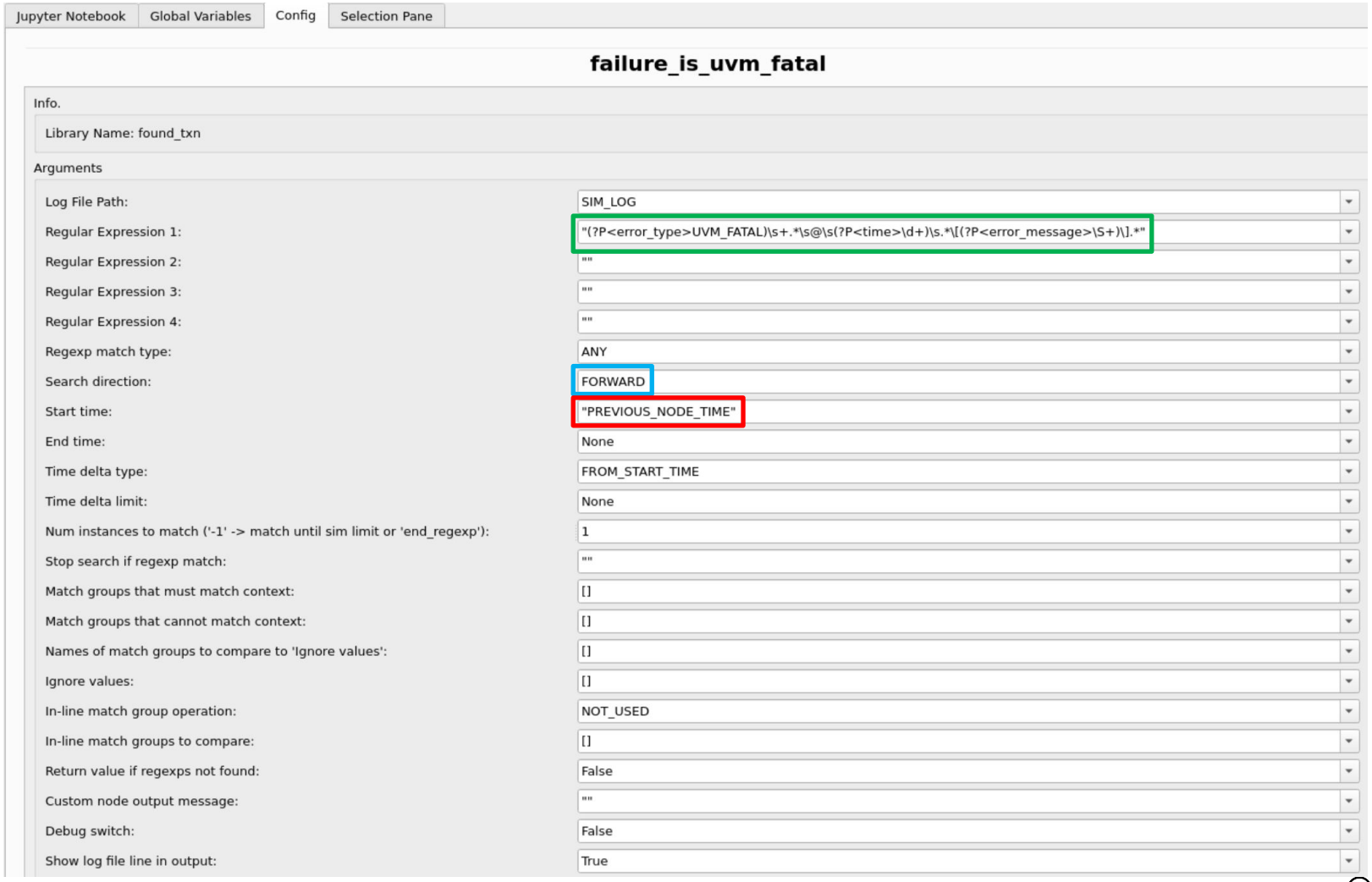


Defining Branch Nodes

Querying unstructured text files:

"Search FORWARD from PREVIOUS_NODE_TIME for 'Regular Expression 1' "

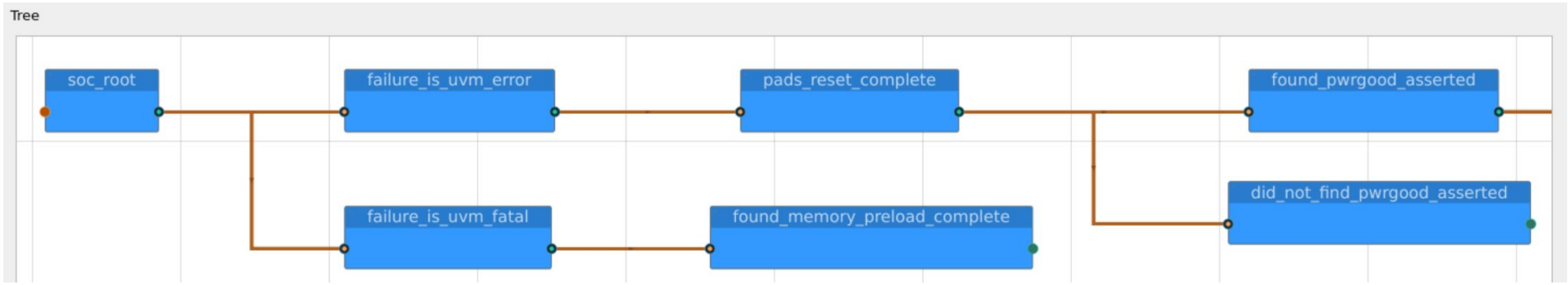
- Match groups are added/updated in context



DDT Tree Output

```
soc_root: TRUE [1]
    failure_is_uvm_error: TRUE [3]
        pads_reset_complete: TRUE [22]
            found_pwrgood_asserted: TRUE [26]
                boot_fsm_did_not_reach_ACTIVE: FALSE [2]
                boot_fsm_reached_ACTIVE: TRUE [15]
                    found_IROM_to_IRAM_transition: TRUE [18]
                        found_ip_wakeup: FALSE [31]
                        did_not_find_ip_wakeup: TRUE [32]
            did_not_find_pwrgood_asserted: FALSE [37]
    failure_is_uvm_fatal: FALSE [34]

Node Output:
[1] []
[3] ['True:  Context check: "status" EQUAL "PASSED" at time 0']
[22] ['[pads_reset_complete] Found signal values at time 995.001: pads_pwrgd_i: h1, pads_reset_i: h1']
[26] ['[found_pwrgood_asserted] Found signal values at time 1195.001: pwrgd_asserted_n_o: h1']
[2] ['[boot_fsm_did_not_reach_ACTIVE] Never found signal values: porfsm_nxtstate: 0101']
[15] ['[boot_fsm_reached_ACTIVE] Found signal values at time 17555.001: porfsm_nxtstate: 0101']
[18] ['[found_IROM_to_IRAM_transition] Found signal values at time 31052.68109: program_counter: h00010000']
[31] []
[32] ['[did_not_find_ip_wakeup] Never found signal values after time 31052.68109: ip_wakeup_i: h1']
[34] []
[37] []
```



Conclusion

1. People are going to be much better at debugging than machines for the foreseeable future.

2. Efficiently capturing and sharing debug knowledge is the problem

3. All debug is a decision tree

4. New failures can be treated as permutations of previous failures enabling reuse of portions of decision tree collateral across disparate failures

5. People interact with data sources in very limited ways that can be abstracted to a tractable set of functions