Cerebus: A Higher-Order Behavior-Based System

Ian Horswill, Northwestern University Computer Science Department
1890 Maple Avenue, Evanston IL, 60201, USA
ian@cs.northwestern.edu

To appear in *AI Magazine*.

Cerebus is an attempt to scale behavior-based robots directly to higher-level cognitive tasks without adjoining a traditional planning system. Cerebus combines a set of behavior-based sensory-motor systems with a marker-passing semantic network, a simple parser, and an inference network, to form an integrated system that can both perform tasks and answer questions about its own ability to perform those tasks. For the 2001 AAAI/IJCAI Robotics Exhibition, Cerebus' demonstration task was to give a formal technical talk on itself, complete with PowerPoint slides and interactive demos.

Cerebus is structured as a parallel network of logic gates and finite-state machines. Inference rules in Cerebus are compiled into a feed-forward logic network. This gives Cerebus' knowledge-base *circuit semantics*: the inputs of the network monitor the truth-values of premises as generated by the sensory systems and the outputs of the network track the truth-values of conclusions in real-time as the premises change. In effect, the entire rule base is rerun from scratch to deductive closure at sensory frame-rates. Although this sounds inefficient, the Cerebus rule engine can run a base of 1000 Horn rules with 10 conjuncts each, updating at 100Hz (100 complete reevaluations of the KB per second), using less that 1% of the CPU. Using a generalization of deictic representation called *role passing*, the network is able to implement a limited form of quantified inference – a problem for previous behavior-based systems. Rules may be quantified over the set of objects in short-term memory, provided they are restricted to unary predicates (predicates of one argument).

Cerebus implements reflective knowledge – knowledge of its own structure and capabilities – through two mechanisms: a marker-passing semantic network provides a simple mechanism for long-term declarative memory, while role passing allows variables within inference rules to be bound to behaviors and signals within the system. The former allows the system to answer questions about its own capabilities, while the latter allows it to answer questions about its current state and control processes.

Cerebus can follow simple textual instructions. When a human types a command like "drive until the turn," its simple parser, which is formed as a cascade of simple finite-state machines, examines each individual word as it is typed, binding the appropriate words to the appropriate roles. In this case, the parser binds the `drive` behavior to the role `activity` and the `turn?` sensory signal to the role `destination`. When it detects a stop (a pause or period), it triggers the `handle-imperative` behavior, which implements the rules:

- If the signal bound to `destination` is false, activate the behavior bound to `activity`.
- If `destination` is bound to a sensory signal and that signal is true, deactivate `activity` and myself.
- If `activity` deactivates itself, also deactivate myself.

Since this behavior is parameterized by other behaviors, we call it a *higher-order behavior*, in analogy to the higher-order procedures of functional programming languages. Other examples are the `explain` behavior, which walks the subtree of the semantic network that describes its argument behavior, to produce a natural language explanation of the behavior, and the `demo` behavior, which both explains and runs the behavior. Role passing and higher-order behaviors are easily implemented using parallel networks of gates and finite-state machines, making them a natural choice for the kind of distributed, parallel processing environments often found on mobile robots. They are implemented in GRL, a functional programming language for behavior-based systems that provides many of the amenities of LISP while statically compiling programs to a network of parallel finite-state machines.

When Cerebus gives a talk, it uses a COM interface to open the specified PowerPoint presentation, then reads the text of each bullet-point and keyword matches it to an appropriate node in its semantic network. It uses a novel distributed representation of a discourse stack to resolve ambiguities using only SIMD marker-passing operations. Having determined the node to which the bullet point refers, it uses spreading activation to mark the subtree rooted at the selected node as being relevant. It then improvises banter about the topic by continually selecting and explaining the "highest priority" relevant, unexplained, node. Priorities are computed off line using a topological sort so that if topic A is required to understand topic B, A will always have higher priority.

By continually reselecting the highest priority relevant unexplained node using circuit semantics, Cerebus can respond instantly to changes in relevance when, for example, an unexpected contingency during a demonstration opens up an opportunity to explain a feature. It also allows Cerebus to cleanly respond to and return from interruptions without replanning. However, such topic shifts require the generation of transition cues such as "but first …" or "getting back to …". Cerebus detects these abrupt topic shifts by tracking the current semantic net node, its parent node, and the previous node and parent. By comparing these, Cerebus can determine whether it has moved locally up, down, or laterally in the hierarchy, or whether it has made a non-local jump to an unrelated node. It then generates the appropriate transition phrase.

Cerebus is far from fluent. It is not intended to demonstrate that behavior-based systems should be the implementation technique of choice for natural language generation. Instead, it shows that parallel, finite-state networks are much more powerful than previously believed. Moreover, by implementing as much of a robot's control program as possible with these techniques, we get efficiency, easy parallelization, and flawless synchronization of the knowledge base with the environment. And all for free.