CHAIMS LANGUAGE

The CHAIMS language is a simple, untyped language designed for composition. It consists of the six primitives shown above as well as functionality for *if* and *case* statements. The traditional call statement is split into these six primitives to provide greater flexibility for the CHAIMS compiler in organizing execution schedules.

SETUP: The SETUP call establishes a connection between the megaprogram and desired megamodule. This includes any initialization of the megamodule that is required. The megaprogrammer may establish default parameters if desired. Unique identity is preserved by giving each SETUP of a megamodule a unique *megamodule handle*.

ESTIMATE: The ESTIMATE call provides a means of avoiding the traditional compiling/execution boundary by doing a runtime estimation of the resources needed to execute a particular invocation. It also avoids the black box problem of object oriented programming.

INVOKE: The INVOKE call starts the execution of a desired application function within a megamodule. This is an asynchronous call as control is immediately returned to the executing megaprogram; the CHAIMS executable does not block. State is maintained by giving each invocation a unique *invocation handle*.

EXAMINE: The EXAMINE call is closely tied to the EXTRACT call. EXAMINE checks the state of a running invocation using the *invocation handle* to see if there is any prepared data to be returned to the megaprogram via the EXTRACT call.

EXTRACT: The EXTRACT call returns data prepared by the invocation to the address space of the CHAIMS megaprogram. Data is packaged together, but individual data within the package may be used by name.

TERMINATE: TERMINATE is used to end a connection between the CHAIMS megaprogram and a megamodule. If it is called using the *megamodule handle,* all invocations to that megamodule are killed. TERMINATE can also be used to end a particular invocation if called using the *invocation handle,* in which case only that invocation is killed.