CHAIMS ARCHITECTURE

The architecture of CHAIMS is designed to facilitate the composition and reuse of heterogeneous services. The architecture can be loosely divided into three portions:

•*Service Modules*

The service modules available for use exist independently of CHAIMS. Some are in existence already, motivated by the need to support legacy code. Others will be developed, some with knowledge of CHAIMS (and thus have native CHAIMS primitives) and some with no knowledge of CHAIMS. Wrapper services will provide translation between the CHAIMS primitives as specified by the megaprogrammer and services which do not support native CHAIMS primitives. We refer to a service as a *megamodule*, as we expect these services to be large, comprehensive, typically remote, and operating autonomously.

• *Distribution Layer*

The distribution layer, which is used to convey the execution sequence created by the CHAIMS compiler to the desired megamodules, can be one of many possible distribution protocols developed for client-server systems such as CORBA, DCE, DCOM, RMI, etc. Any distribution protocol can be used, so long as there is a runtime library which allows the CHAIMS compiler to create the appropriate interface to the distribution layer. The distribution layer is invisible to the megaprogrammer but is the element of the architecture most critical to distributed computing and thus forms a major component of the CHAIMS architecture.

• *Application Megaprogram Written in CHAIMS*

The CHAIMS compiler takes the program specified by the megaprogrammer and generates an appropriate execution sequence which interfaces to whichever distribution layer is being used. The compiler optimizes the execution sequence as much as possible.