### Title and Subtitle
Flexible Software Composition

### Authors
Gio Wiederhold and Neal Sample

### Performing Organization Name(s) and Address(es)
Office of Sponsored Research
Stanford University
Stanford, CA 94305

### Sponsoring / Monitoring Agency Name(s) and Address(es)
AFOSR/IM
110 Duncan Ave, Room B115
Bolling AFB, DC 20332

### Abstract
The CHAIMS effort is developing and refining a high-level programming language for software module composition. In the last two years, that language has been codified and is known as CLAM, the Composition Language for Autonomous Megamodules. The language is currently supported by the CPAM (CHAIMS Protocols for Autonomous Megamodules) runtime system. CPAM is the interface between the compiled megaprogram (client) and the megamodules (servers).
The CHAIMS effort is developing and refining a very high level programming language for software module composition. In the last two years, that language has been codified and is known as CLAM, the Composition Language for Autonomous Megamodules. The language is currently supported by the CPAM (CHAIMS Protocols for Autonomous Megamodules) runtime system. CPAM is the interface between the compiled megaprogram (client) and the megamodules (servers).

The CHAIMS compiler generates a variety of invocation sequences for current and developing standards for software interoperation, with a focus on multi-computer, distributed operation. The language includes the ability to set up module interfaces prior to executions, request performance estimates from modules prior to their invocation, schedule modules in parallel, monitor execution of invoked modules, interrupt inadequately performing modules, and provide data and meta-information to customer interface modules.

CHAIMS supports a paradigm shift that is already occurring: a move from coding as the focus of programming to a focus on composition. This shift is occurring invisibly to many enterprises, since there is no clear boundary in moving from subroutine usage to remote service invocation. There are hence few other tools and inadequate education to deal with this change.

**CHAIMS Literature:**

Sample, Beringer, Wiederhold: A Comprehensive Model for Arbitrary Result Extraction; awaiting publication

Wiederhold, Beringer, Sample, Melloul: Composition of Multi-site Services; accepted for IDPT'99, Kusadasi, Turkey, June 1999 (conference was moved to IDPT 2000, Dallas, USA)

Beringer, Wiederhold, Melloul: A Reuse and Composition Protocol for Services; SSR'99, colocated with ICS'99, Los Angeles, May 1999

Melloul, Beringer, Sample, Wiederhold: CPAM, a Protocol for Software Composition: CAiSE'99, Heidelberg, Germany, June 1999; Springer LNCS volume 1626

Sample, Beringer, Melloul, Wiederhold: CLAM: Composition Language for Autonomous Megamodules; Third Int'l Conference on Coordination Models and Languages, COORD'99, Amsterdam, April 26-28, 1999


Beringer, Tornabene, Jain, Wiederhold: A Language and System for Composing Autonomous, Heterogeneous and Distributed Megamodules; DEXA International Workshop on Large-Scale Software Composition, Vienna Austria, August 28, 1998

Tornabene, Jain, Wiederhold: Software for Composition: CHAIMS; position paper for the Workshop on Compositional Software Architectures of OMG, DARPA and MCC in Monterey, CA, January 6-8, 1998

Perrochon, Wiederhold, Buback: A Compiler for Composition: CHAIMS; Fifth International Symposium on Assessment of Software Tools and Technologies (SAST'97), Pittsburgh, June 3-5, 1997

Wiederhold, Wegner, Ceri: Towards Megaprogramming; CACM, Nov.1992