CHAIMS: Mega-programming research

Gio Wiederhold

Stanford University

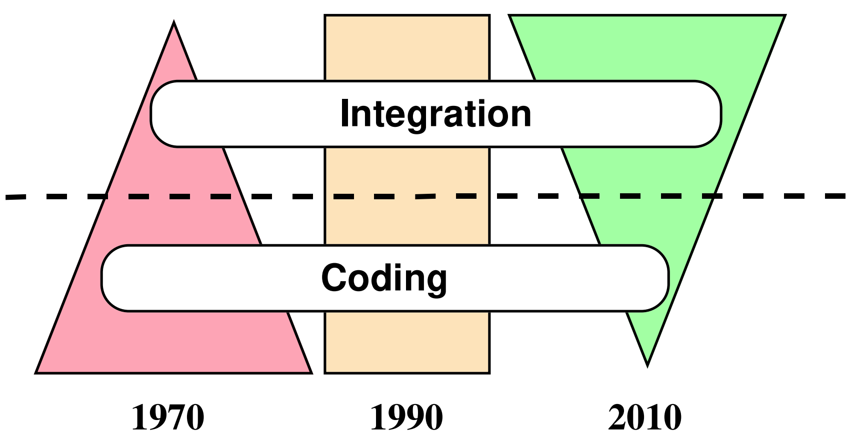
Objective: Investigate revolutionary approaches to large-scale software composition.

Approach: Develop and validate a composition - only language, exploiting existing client-server interface conventions.

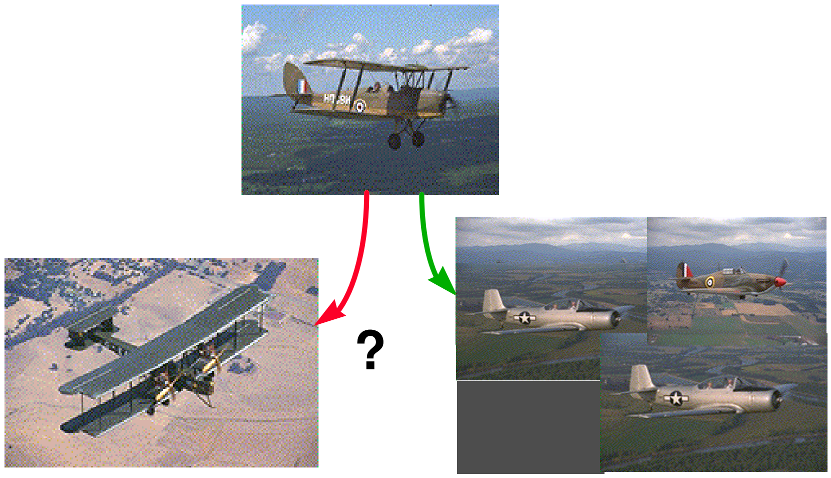
Planned contributions:

- Hardware and software platform independence.
- Clear and executable software architecture.
- Potential for multi-site dataflow optimization.

Shift in Programming Tasks

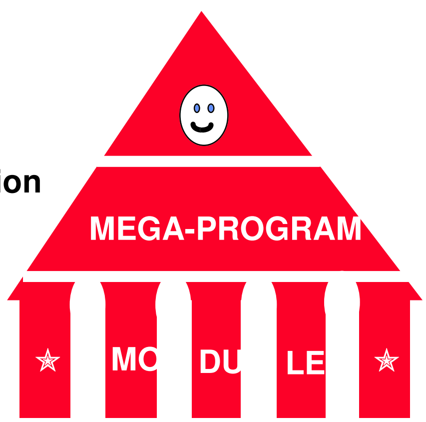


Scaling alternatives

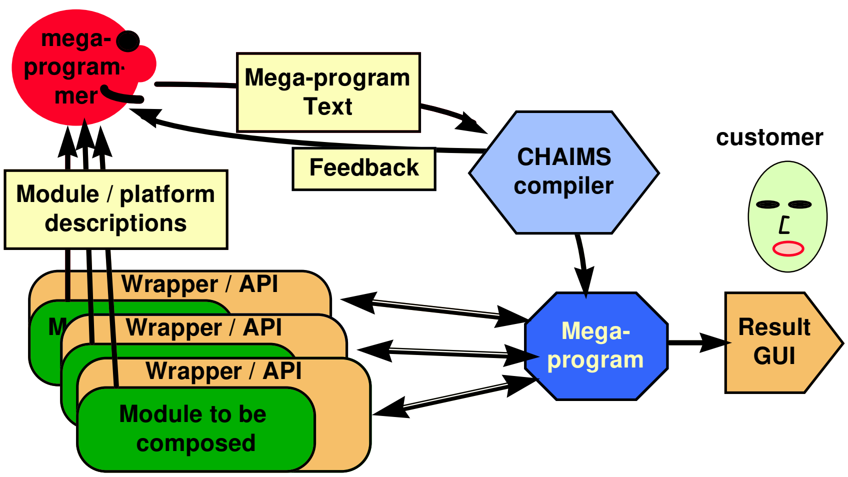


CHAIMS principal components

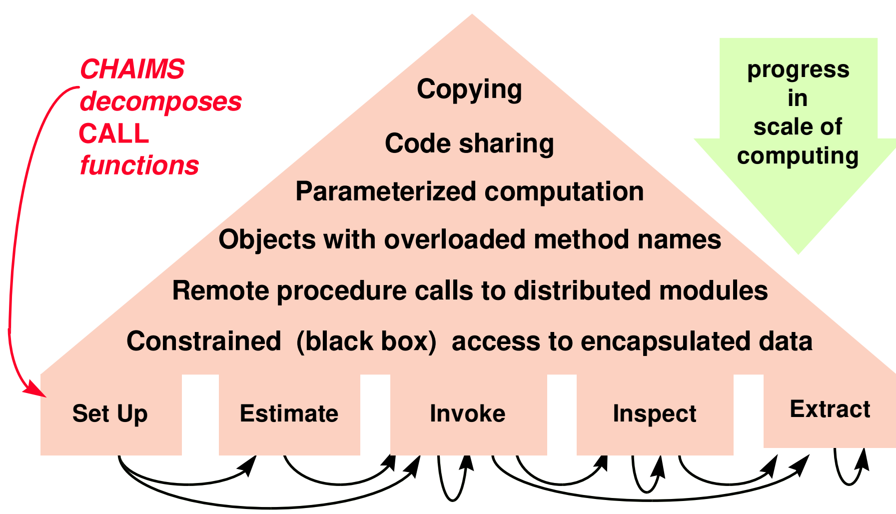
- **Mega-programmer: defines architecture**
- **Mega-program: schedules the invocation of modules**
- **Standard client-server interfaces**
- **Modules: preexisting and written to order**



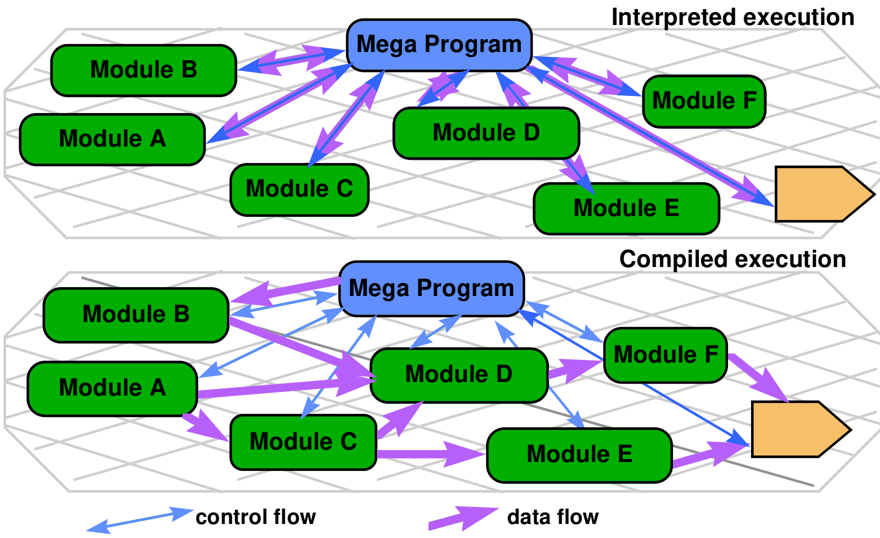
Mega-programming Process



Decomposing CALL statements



Compiling into a Network

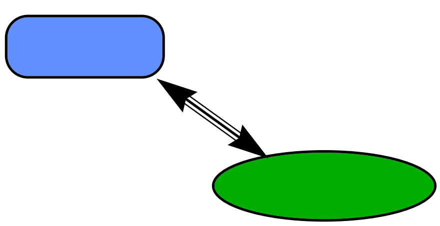


Generate interface languages

Being developed for Client-Server systems

specify representations and protocols

- **Common Object Broker: CORBA with IDL**
- **Microsoft OLE**
- **Distributed Computing Environment DCE**
- **Knowledge Query & Manipulation Lang. KQML**



Paying for SW Services

You can not run an effective (SW) business and not be reimbursed for it. How? Four approaches:

- **Buy Software** *buy oilfield*
- **Lease copy / usage rights** *lease well*
- **Time / user limited access** *fill tank*
- **Pay by use instance** *take bus*

General problems, differ by approach

- **copy protection?** *B - -, L -, T+, P++*
- **keeping SW updated** *B - -, L -, T++, P++*
- **accounting and billing** *B++, L+, T -, P - -*
- **performance** *B++, L+, T -, P - -*

Chaims project tasks

- **Analyze commonalities, differences in interface capabilities of client-server languages**
- **Define essential subset as CHAIMS functionality**
- **Specify minimal language**
  - **steal syntax (Ada?)**
  - **minimal typing {Integer, Ids, bags, ...?}**
  - **avoid type conversion**
- **Implement API subset for CORBA, DCE usage**
- **Wrap / construct several programs for demos**
- **Demonstrate use in heterogeneous setting**
- **Define full-scale demonstration**

Questions to be answered

- **Is a Megaprogramming language focusing on composition feasible?**
- **Can it exploit on-going progress in client-server models?**
- **Can it express clearly a high-level distributed SW (*DSS*) architecture?**
- **Can it affect SW process concepts and practice?**
- **Can high-level dataflow among distributed modules be optimized?**

CHAIMS