



---

**Daniel N. Hennessy**

Lockheed AI Center  
Senior Scientist

---



---

# Prototyping Environments

Prototyping in MACL

---

# What I will Discuss

---

- What is prototyping?
- Why would someone use Lisp?
- Why would someone use Macintosh Allegro Common Lisp?

# What Makes a Prototype Successful?

---

- Proof of concept
- Graphics
- Flexible
  - Modular
  - Structured
  - OOP
- Inexpensive

# A Prototype Software System

---

- Does:
  - Simulate the important interfaces
  - Performs the main functions of the intended system
- Does not:
  - Handle unusual cases
  - Respond correctly to invalid inputs
  - Abort cleanly

# Computer Language Trade-Off

---

- Efficiency
  - Memory efficiency
  - Speed efficiency
- Expressiveness
  - Number of pre-defined functions
  - Naturalness of syntax
  - Release from low-level functionality

# Lisp Advantages

---

- Expressiveness
  - Many functions
  - Natural syntax
- OOP
- Release from low-level functionality
  - Data typing
  - Memory management

# Lisp Problems

---

- Cost
- Stability
  - Dialects and portability
- Efficiency
  - Memory and speed
- Debugging capabilities
- Deliverability
- System integration



# MACL as a Prototyping Language

---

- Well integrated with Mac environment
- Utilities
  - Foreign function interface
  - Allegro Interface Designer (AID)
- Inexpensive
- Stable/Portable
  - Common Lisp and CLOS
- Deliverable

# MACL Problems

---

- Efficiency
  - Memory
  - But larger applications
- Debugging capabilities
  - Traces
  - Continuation from errors

# Why Should I Program in MACL?

---

- Programming ease/Expressiveness
- Modularity/OOP support
- System integration/Graphics
- Inexpensive

# Clavier

---

- Interfaces
  - Rapidly prototyped in MACL and HyperCard
  - Re-implemented in MACL
- Algorithms
  - Prototyped and revised in MACL
- Effort
  - 20,000 lines of code
  - 14 person-months



The power to be your best