Challenges for Logic Programming

Steve Miner

@miner
steveminer@gmail.com

Clojure Conj 2012

Prologue
Last year at the Conj

- Dan Friedman and William Byrd - miniKanren
- Ambrose Bonnaire-Sergeant - core.logic (ported by David Nolen)
- Jim Duey - recent fork/join work

Outline

- Personal experience
- Historical narrative
- Boring filler
- Dramatic conflict
- Unsatisfying ending

Challenge

- An objection or query as to the truth of something, often with an implicit demand for proof

- Why learn logic programming?

25 years ago

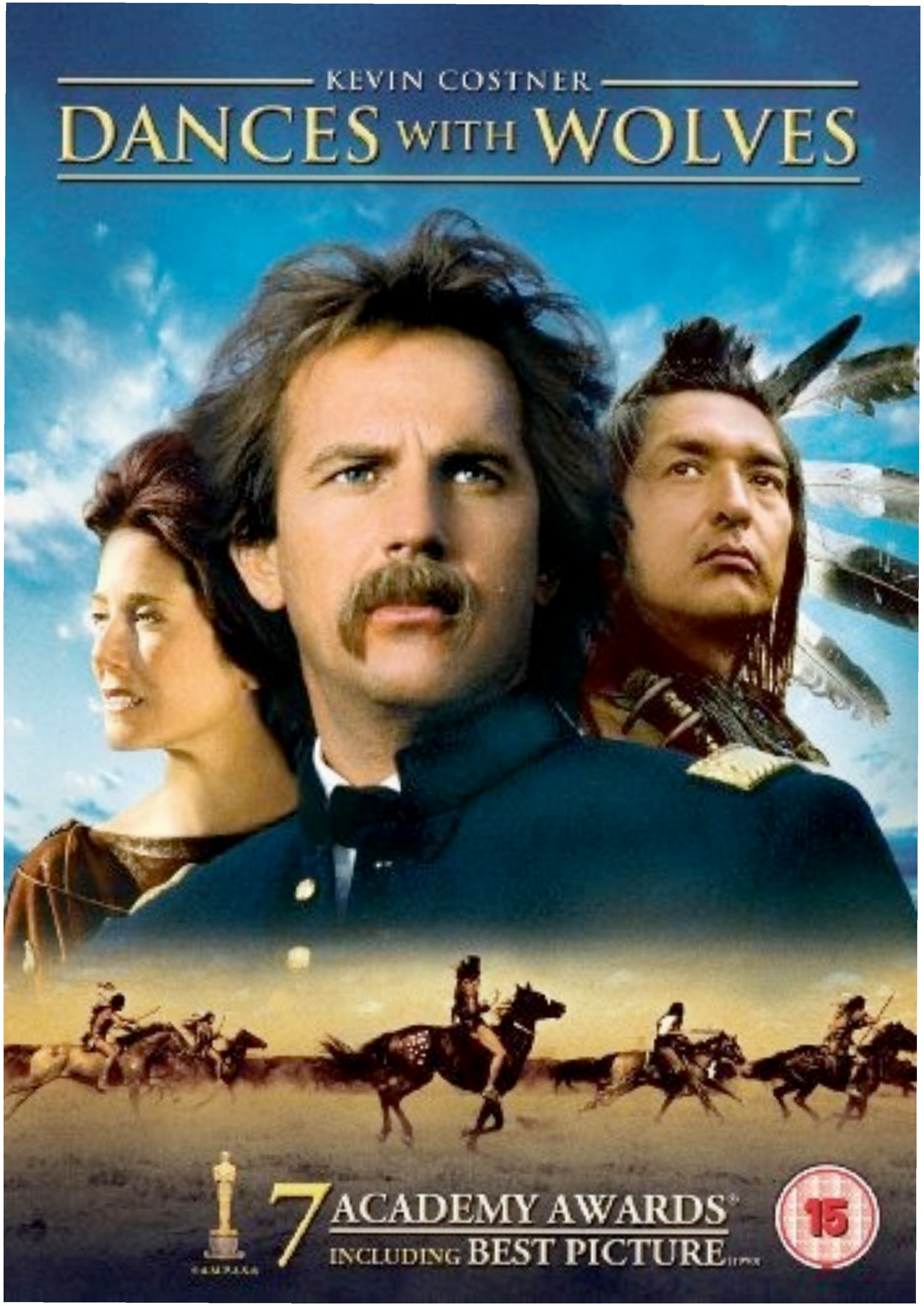
- A Lisp programmer
- Expert systems and planning
- Discovered Prolog

AALPS

- Automated Air Load Planning System
- Loadmaster: logistics, restraints, weight, balance
- Army-wide standard for cargo-aircraft load planning and analysis
- small team of Prolog programmers at SRI

Cultural Challenge

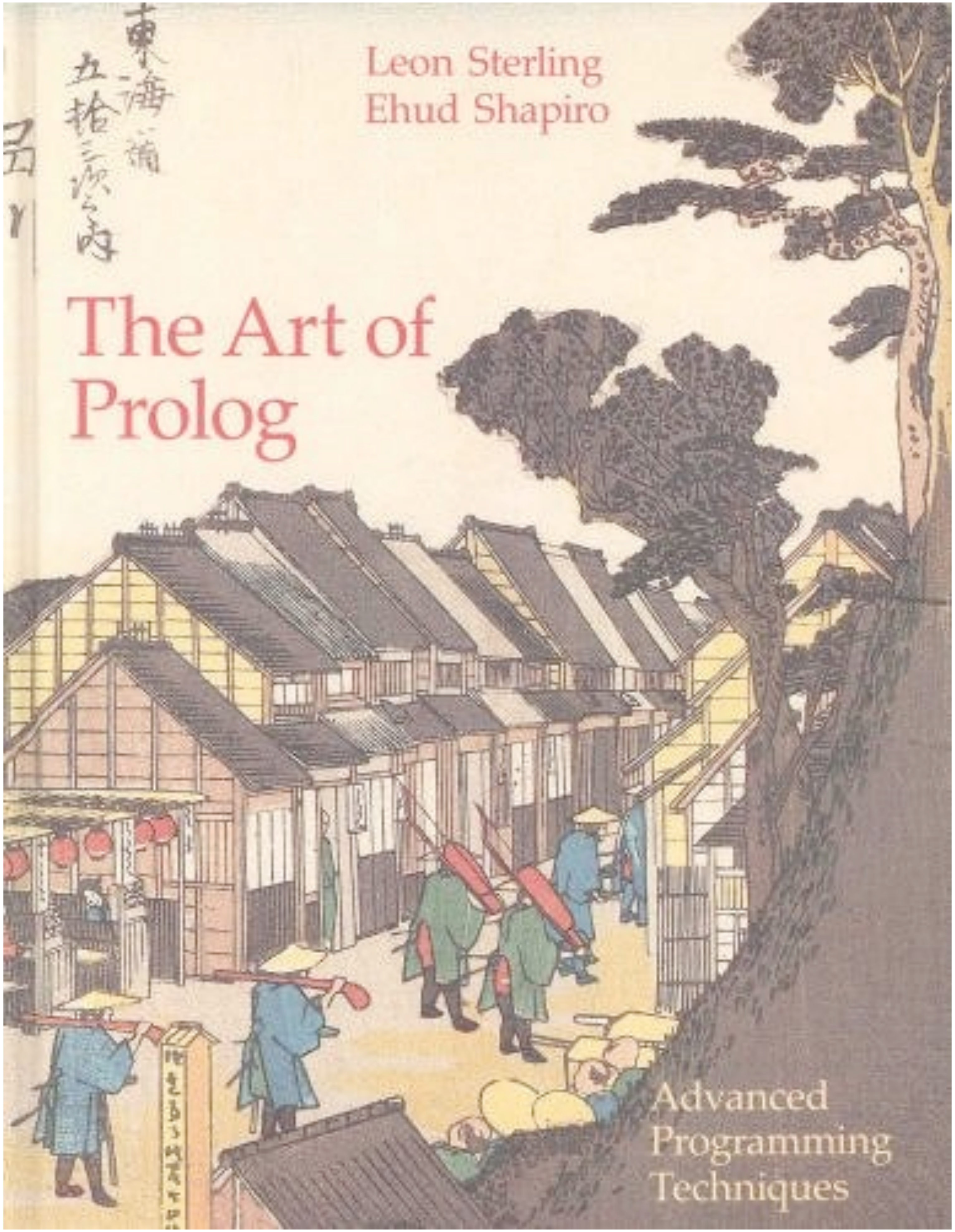
- Common Lisp
  - s-exprs, macros, CLOS
  - full control
- Prolog
  - strange syntax
  - unification and backtracking


KEVIN COSTNER
DANCES WITH WOLVES
7 ACADEMY AWARDS®
INCLUDING BEST PICTURE
15

Logic Programming

- inspired by logic
- facts, rules, queries
- defining relationships between objects
- computation is deduction

Pearl Fryar



Leon Sterling
Ehud Shapiro

The Art of Prolog

Advanced Programming Techniques

Art of Prolog

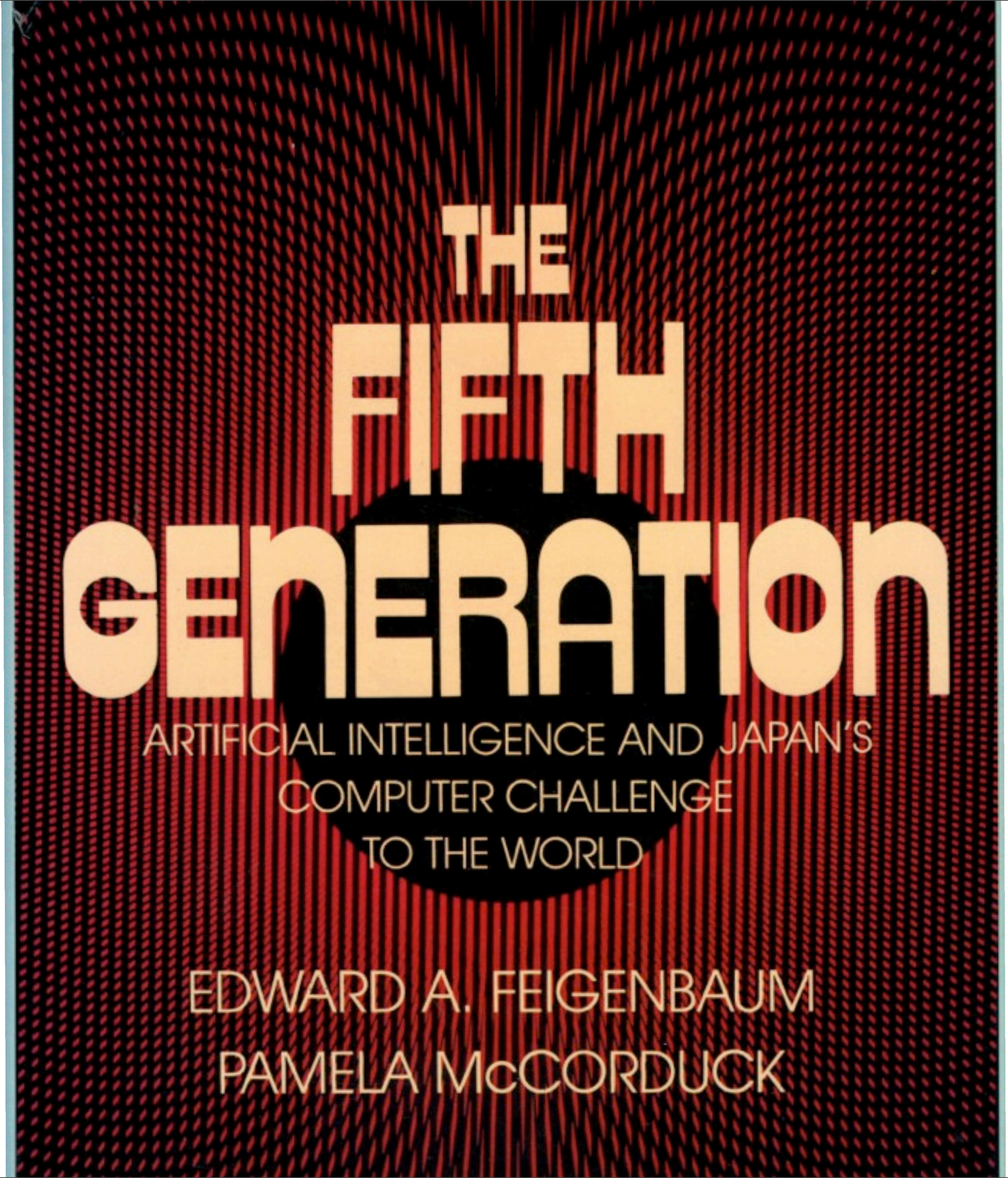
- Leon Sterling and Ehud Shapiro, 1986
- “In Prolog programming (in contrast, perhaps, to life in general) our goal is to fail as quickly as possible”
- Shapiro: Concurrent Prolog
- consulted on Fifth Generation Project

Benefits

- Precise problem statement
- Elegance
- Declarative semantics
- Procedural execution
- Flexible and transformable
- Verification by proof (maybe)

Don Knuth

- “Beware of bugs in the above code; I have only proved it correct, not tried it.”



THE FIFTH GENERATION

ARTIFICIAL INTELLIGENCE AND JAPAN'S
COMPUTER CHALLENGE
TO THE WORLD

EDWARD A. FEIGENBAUM
PAMELA McCORDUCK

The Fifth Generation

- Artificial Intelligence and Japan's Computer Challenge to the World
- Edward A. Feigenbaum and Pamela McCorduck, 1983

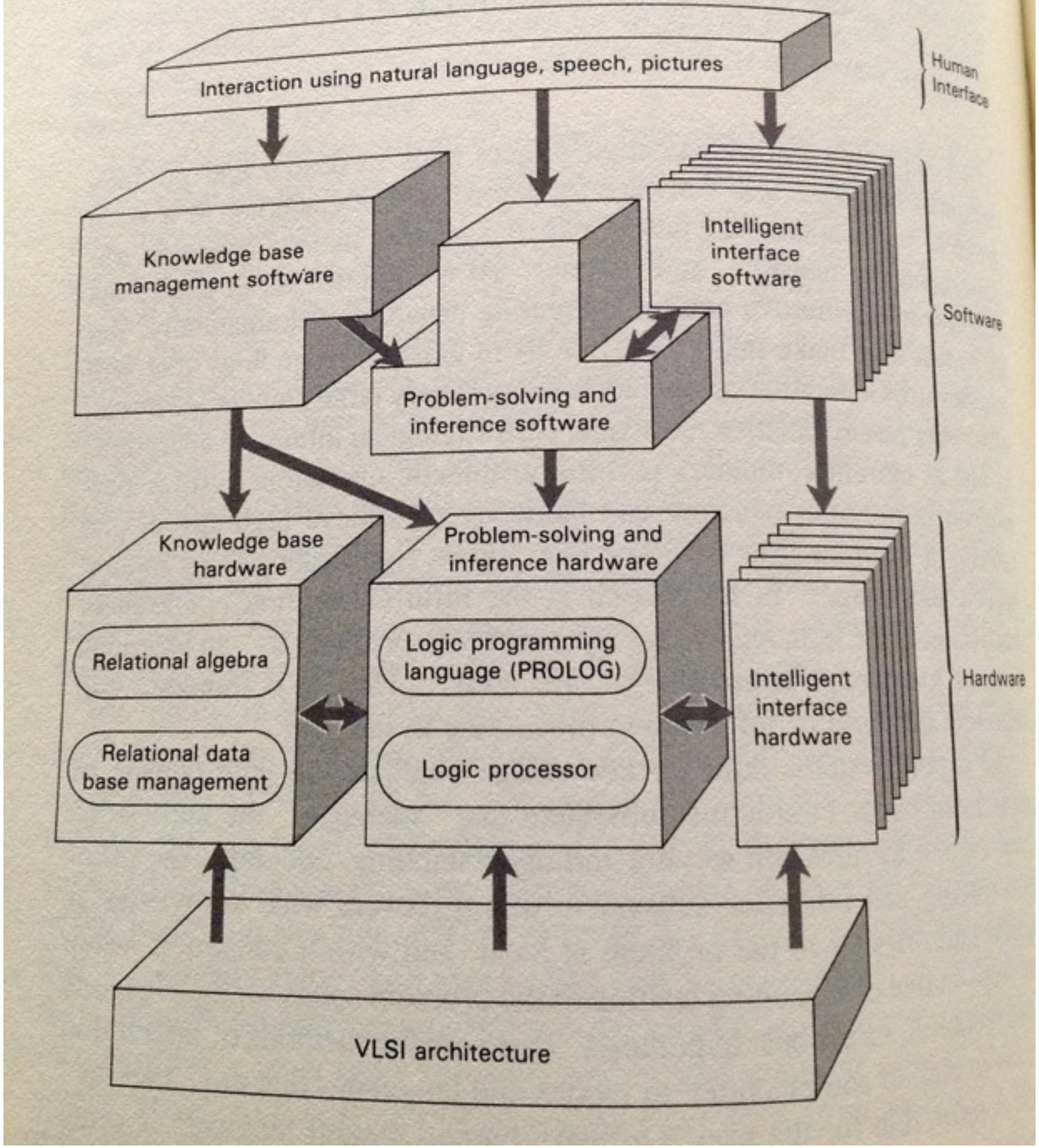
Edward Feigenbaum

- Knowledge Systems Lab at Stanford
- 1994 ACM Turing Award for AI work
- IntelliCorp and Teknowledge

The Fifth Generation

- Japanese Government sponsored (MITI)
- Parallel hardware
- Prolog based software
- Natural language interface
- Knowledge Information Processing Systems (KIPS)

Fifth-Generation Computer Systems


Interaction using natural language, speech, pictures
Human Interface
Knowledge base management software
Problem-solving and inference software
Intelligent interface software
Software
Knowledge base hardware
Relational algebra
Relational data base management
Problem-solving and inference hardware
Logic programming language (PROLOG)
Logic processor
Intelligent interface hardware
Hardware
VLSI architecture

Mixed reactions

- Feigenbaum saw US falling behind
- IBM didn’t care
- Lisp hackers were skeptical
- Prolog proponents anticipated success
- Europe had cut back on funding CS
- DARPA - AI Winter is coming

"Development of Logic Programming"

- paper by Carl Hewitt, 2008 (many versions)
- What went wrong
- What was done about it
- What it might mean for the future

Carl Hewitt

- Planner language (1969)
- procedural embedding of knowledge
- Actor model of computation
- Direct Logic (TM) to deal with inconsistency
- now visiting professor at Stanford

Hewitt on Prolog

- “Prolog was basically a subset of Planner that restricted programs to clausal form using backward chaining and consequently had a simpler more uniform syntax.”

"The birth of Prolog"

- Alain Colmerauer on Planner:
- “The lack of formalization of this language, our ignorance of Lisp and, above all, the fact that we were absolutely devoted to logic meant that this work had little influence on our later research.”

Hewitt on Planner

- backtracking proved to be too rigid and uncontrollable
- single global data base which was not modular or scalable
- the syntax was not a pretty sight
- Prolog is a more controlled approach

What Went Wrong

- Clausal form hides the underlying structure of the information
- Practical domains of knowledge are inconsistent
- Proof by contradiction is not a sound rule of inference for inconsistent systems

Japanese Fifth Generation

- Hewitt quoting Robert Kowalski:
- Logic Programming was virtually unknown (1980s)
- Most of its research activity was in Europe.
- Logic programming was to play a central, unifying role in the FGCS Project.

Unifying Role of LP

- Kowalski: “computation could be subsumed by deduction”
- Pat Hayes: “Computation = controlled deduction”
- Hewitt disagrees
- Fifth Generation as a test case

Failure of Fifth Gen

- High risk approach
- Consortium wasn’t totally committed
- Hardware was late
- Natural language interaction wasn’t ready
- Parallelization and committed choice
- Prolog took the blame

Logic inspired

- languages: Concurrent Prolog, Mercury, Oz, λProlog, Shen, miniKanren, core.logic
- Constraint Logic Programming
- SAT solvers
- Rule-based systems (OPS5, CLIPS, Drools)
- Datalog

Haskell

- Phillip Wadler on Type Inference
- "Faith, Evolution, and Programming Languages" on YouTube
- Simon Peyton Jones on Data Parallelism
- "Harnessing the Multicores" on YouTube

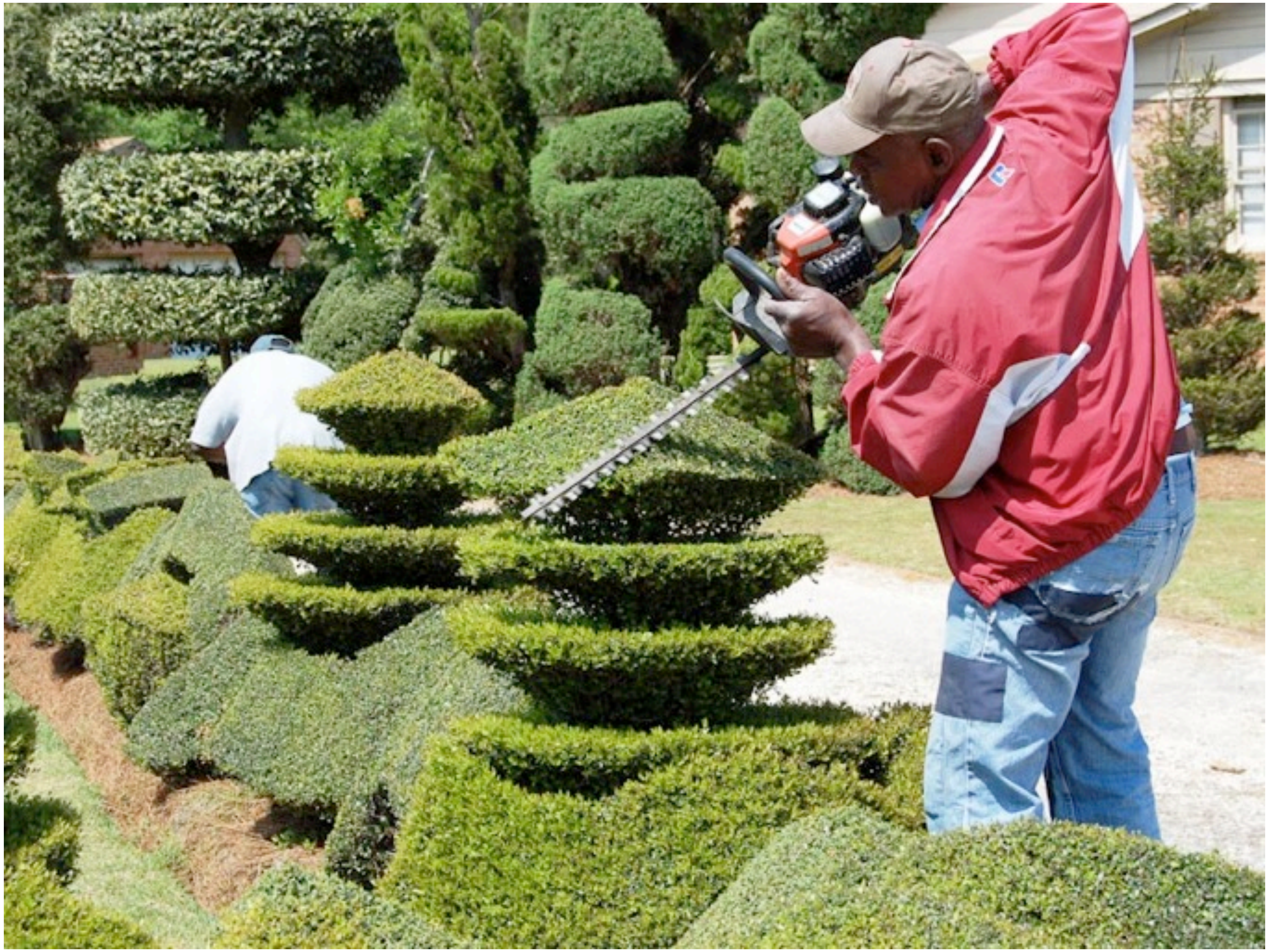
Clojure can help

- Community
- Functional Programming is a good host
- Facts, Rules, even Models can be values
- Run everywhere + the browser
- Concurrency

Meeting Challenges

- Awareness
- Prove usefulness to engineers
- Beyond toy problems
- Clojure adoption

Pearl Fryar



References

- Hewitt - http://www.aaai.org/Papers/Workshops/2008/WS-08-14/WS08-14-003.pdf
- Colmerauer - http://reference.kfupm.edu.sa/content/b/i/the_birth_of_prolog__24094.pdf
- Feigenbaum - http://ksl-web.stanford.edu/people/eaf/
- http://pearlfryar.com
- Anderson, D. and Ortiz, C.L. "AALPS: A Knowledge-Based System for Aircraft Loading" IEEE Expert, Winter 1987.

The End

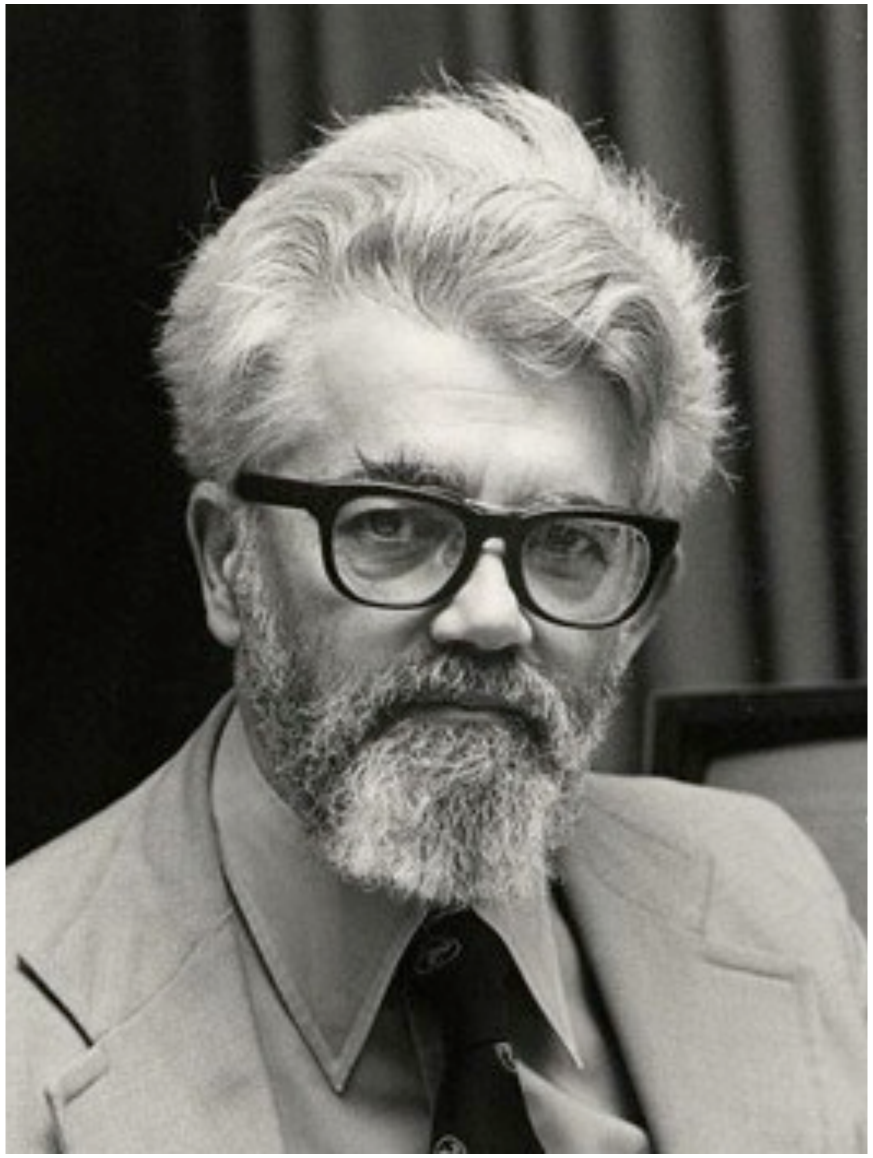
- Steve Miner
- @miner

Extras

bonus slides

John McCarthy

- Father of Lisp
- Logical approach to AI
- Nonmonotonic reasoning
- http://www-formal.stanford.edu/jmc/



Frame Problem

- Minksy's "A Framework for Representing Knowledge"
- What stays the same after an action?
- Default reasoning, unless abnormal
- Circumscription based on minimal models

Research work

- McCarthy’s context for situational calculus
- Meta-level reasoning
- Generating rules
- Data models or models as data

Edsger Dijkstra

- “The required techniques of effective reasoning are pretty formal, but as long as programming is done by people that don't master them, the software crisis will remain with us and will be considered an incurable disease. And you know what incurable diseases do: they invite the quacks and charlatans in, who in this case take the form of Software Engineering gurus.”

H.L. Mencken

- “Philosophy first constructs a scheme of happiness and then tries to fit the world to it.”