HPC Tradecraft for Computer Scientists: What We Stopped Teaching

Edward Barnard

HPC Tradecraft for Computer Scientists: What We Stopped Teaching

HPC Tradecraft Apprenticeship, Volume 1

Edward W. Barnard

This book is available at https://leanpub.com/tradecraft

This version was published on 2026-04-04

Leanpub

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Also By Edward W. Barnard

Teaching Mastery in the Classroom
Preparing to Survive the Early-Career Power Imbalance
Francis Barnard Part A
Teaching Mastery in the Classroom
Large Language Model Architecture Patterns in PHP: No Mathematics Required
The Impossible Challenge Manual for Age 14 and Up, Even For Adults: How to accomplish what everyone says you can't
The Wizard's Lens: Learn to Think Like AI
Nobody but Us: A History of Cray Research's Software and the Building of the World's Fastest Supercomputer
Beyond Prompt Engineering
Billy Mitchell's Bombsight: Shaping the B-25 Mitchell Bomber

Contents

Chapter 1. When Performance is Everything **1**
Three Questions 1
Improved Efficiency 2
New Capability 3
Constraint-Based Design 5
Tradecraft Transmission 6

Chapter 2. Apprenticeship Transmission **7**
Code Generation as Example 7
Visibility Lost 7
Transmission Method 7
Book Design 7
Conditions for Formation of Thinking 7
Example Usage 7

Chapter 3. Teaching and Creating Mastery **9**
Road Versus Map 9
Parallel and Equivalent Routes 9
Classroom Practice 10
Mental Models 10
Three-Step Process 11
Mental Models Enable Analogies 11

Chapter 4. Transcendent Patterns: Teaching the Process of High-Tech Mastery in Student-Accessible Fashion **12**
Definition 12
Fun and Superpower as Motivation 12
Tools of Transmission 12
Three-Step Process 12
Transcendent Patterns 13
Practicing the Insight 13

Transcendent Insight . . . 14
Teaching This Skill . . . 14
Cognitive Design of This Chapter . . . 14
Summary . . . 15

Chapter 5. If You Want More . . . 16
The Barrier . . . 16
Capabilities . . . 16
Solutions . . . 16
Cognitive Design of This Book . . . 16

Declassified Documents and Road Map . . 18

Appendix A. Analytical Machine Employment (1952) . . . 19
Purpose of Analytical Machines . . . 19
Machines according to Potentiality . . . 19
Employment Related to Potentiality . . . 19
Present Employment Versus Reserve Capacity . . . 19
Employment of the Revolutionizers . . . 19
Employment of the Labor-Savers . . . 19
IBM's Labor-Saving Employment . . . 20
The ROBIN Job . . . 20
ROBIN Utilization . . . 21
ROBIN Preparation . . . 21
GOLDBERG . . . 21
Miscellaneous Small Operations . . . 21

Appendix B. Cryptanalytic Machines in NSA (May 1953) . . . 22
Dr. Howard Campaigne . . . 22
Bombes, Scritchers, and GOLDBERG . . . 22

Appendix C. HPC Tradecraft Road Map . . . 23
The 1995 Barrier . . . 23
Two Series . . . 24
Book 1. HPC Tradecraft for Computer Scientists: What We Stopped Teaching . . . 24
Book 2. Nobody but Us: A History of Cray Research and the Building of the World's Fastest Supercomputer . . . 25
Book 3. The Wizard's Lens: Learn to Think Like AI . . . 25

Book 4. High-Stakes Ethics 25
Book 5. Unexpected Histories: Spotting Patterns and Making Connections That Others Miss 26
Book 6. Constraint-Based Design: A Gateway to AI 26

Chapter 1. When Performance is Everything

What was so unusual and special about Cray Research? We accomplished, repeatedly, what nobody else on the planet was able to match. We embodied a high-stakes environment verified by decades of operational success.

High-Performance Computing (HPC) is **not** merely a branch of computer science characterized by parallelism and large-scale systems. That is the public-facing decoy version. HPC characterizes the problem space where wrong or insufficient decisions mean large numbers of people might die.

My role is to transmit our tradecraft as practiced within Cray Research **and** to transmit it within the context of our high-stakes operational environment. As part of that transmission, we will demonstrate this tradecraft using modern AI models **as HPC systems** to show our methods remain important today.

Now that I have (perhaps shockingly) told you what HPC is *not*, I can show you what it *is*.

Three Questions

High-Performance Computing organized around three questions:

1. What, or how much, improvement in computing system performance, is needed to enable new capabilities currently impossible to consider?
2. Given a machine with dramatically improved performance, what can we get this machine to do that has been impossible to accomplish, even in part?
3. What problem could save many lives if we could solve it, but solving requires a capability we do not have? What must that enabling capability look like? (We always treat this as a two-step question because the customer asks the first part and we as the vendor answer the second.)

At Cray Research we focused on the first question: create computing system capabilities that enabled solutions not currently possible, even in part. Our customers, both classified and non-classified, came to us because of question two. We knew the third question was our reason for existence, and knew both question and answer were outside our reach.

Handling that third question sounds contradictory. That is the nature of working alongside classified environments. Hypothetically, we might propose a specific performance capability and ask if that is sufficient. The customer can answer yes or no without disclosing the reason for needing that capability. Collaboration then becomes possible in terms of that stated computing system performance, with the *need* for that performance never entering the conversation.

We were thus in the business of building the world's fastest supercomputers without telling our customers what the computing systems were good for. We demonstrated system capability and let the customer decide if they wanted one or not (they wanted one).

The purest example is CRAY-1 serial number 1, the first Cray Research mainframe computer. It had no software, just five tons of bare metal on a loading pallet. The customer knew how to write software; they needed a more powerful computer to run it.

All high-performance computing systems have a bounding constraint. It might be memory size or speed. It might be heat generated. It might be electricity consumption. Liebig's Law of the Minimum is a good analogy: plant growth is limited by the relatively scarcest ingredient. If you overcome or alleviate the bounding constraint, you can achieve greater performance. But you will still hit the next constraint. There will always be a bounding constraint; otherwise, performance would be infinite.

With the second CRAY-1 system, serial number 3 (serial 2 was scrapped for a memory redesign), the constraint was lack of software. The National Center for Atmospheric Research (NCAR) could use a CRAY-1 for improved weather forecasting, but without an operating system and FORTRAN compiler, it was useless to them. The bounding constraint is not always physical or obvious.

Once we have a capability, what do we do with it? The NSA (National Security Agency) explains.

Improved Efficiency

Appendix B is a declassified NSA document describing the HPC process. It is the same situation as modern AI, but 70 years earlier and in a Top Secret setting. The project name was BOOTSTRAPS. This was a cryptanalytic (code breaking) problem.

The NSA initial analysis determined that this problem needed several hundred man hours, with the result far from certain. They estimated the cost as nearly $1,000 (in 1953 dollars). That was impractical. They prioritized other projects and set this one aside. This one was not the best use of available expert analysts.

The NSA had a specific method, which they called a "pass," for solving the BOOTSTRAPS problem. But the manpower requirement was too exorbitant with current resources.

Then someone found a way to automate the pass using card equipment. This is the value of a high-speed computing system. Now the cost for a pass was approximately $32.50. That 30X improvement crossed the feasibility barrier, and the NSA made several hundred passes. But that caused a problem downstream.

The result of a sequence of passes was *not* the final answer. It was material from which the cryptanalysts could proceed to a solution. The new labor-saving method led therefore to more work for cryptanalysts. Each solution opened new jobs to do as well. The new technique (automate the pass) using computing capability (card equipment) created an opportunity not previously feasible. But the result was so valuable that it created more work, rather than less, for the expert analysts.

The 1950s direct predecessor to the 1970s CRAY-1 supercomputer was a computer named ATLAS. When ATLAS became operational, someone programmed ATLAS to process the BOOTSTRAPS pass. The cost of a pass became $1.25. This cost was now such a bargain that all available data was run through the new process, making a tremendous job for the NSA cryptanalysts, and ultimately a tremendous amount of actionable plain text.

With BOOTSTRAPS, the NSA concluded that when analytic machinery enables capabilities not previously feasible, it makes more work for the analyst rather than less. (They undoubtedly rated this a good outcome, but the declassified document does not say.)

New Capability

Consider a second scenario. Suppose the NSA had the necessary dozens of analysts to solve the BOOTSTRAPS problem by hand. The card equipment then allowed that job to be performed about 30 times faster or more efficiently. The (hypothetical) 30 people assigned to that part of the project could be reduced to a single person, freeing up the other 29 for some other task, or enabling a significant workforce reduction.

When ATLAS became operational and available for this project, that represented another 25X reduction in manpower requirement.

This is exactly the approach being promoted by the Big Tech giants: junior developers can be replaced with AI. Greater efficiency means tremendous manpower savings.

But this is not what High-Performance Computing is for. The BOOTSTRAPS analysis shows the difference.

BOOTSTRAPS was a known problem to solve, but impractical to attempt. Automation made it practical, even if just barely. Note that "just barely" meant a 30X speedup, not an incremental improvement such as 10% or 25%.

This is the distinction that we stopped teaching around 1995 (as explained in "Constraint-Based Design" below). HPC tradecraft provides those timeless skills that AI cannot touch:

1. The Cray Research (and its predecessors) engineering tradition that produces unmatched potential.
2. The creativity, more an attitude and orientation than a skill, of devising uses never before considered.
3. Observing, characterizing, and using existing HPC systems (particularly including AI) in ways assumed to be impossible.

The NSA (and predecessor agency AFSA) made a sharp distinction between using computing systems as:

- Labor savers, reducing manpower cost of existing projects and tasks
- Revolutionizers, devising new capabilities previously impossible to consider, even in part

Appendix A is a declassified AFSA document explaining this distinction, placing ATLAS in the “revolutionizer” category based on its performance capability.

Now I can explain what “HPC tradecraft” means in practice.

Constraint-Based Design

High-Performance Computing (HPC) tradecraft consists of constraint-based design. Constraints shape solutions. Similar constraints tend to shape similar solutions.

HPC design is never building a computer and then seeking a product/market fit. Design always addresses a known problem or need. Design always aims to solve a problem that cannot currently be solved.

Abstractions Enable Advancement

These tradecraft facts present a difficulty: since around 1995, we began to hide physical constraints behind abstractions, libraries, and infrastructure. That freed **normal commercial and scientific computing** to rapidly advance. But it shunted HPC tradecraft to the shadows, consisting of tacit knowledge passed from person to person and never written down in the open literature.

Yet down on the bare metal, HPC tradecraft remains constraint-based design. AI Large Language Models (LLMs) are in fact designed as HPC systems.

Constraints Became Hidden

Do you see the barrier? We do not normally reason about constraint-based design because we hid the constraints. The barrier is that simple. And, therefore, easily surmounted, but only if you allow me to show you the way.

HPC tradecraft **always** begins with the constraints. After finding the true bounding constraint, we then measured our actual capabilities, or named what capability must come into existence. Capabilities bounded our solution space.

The next step is simple: accomplish what has never been done before. At Cray Research and our predecessors, we accomplished what nobody else on the planet could, repeatedly.

How did we do that? We spotted or used patterns and made connections that others missed. Once you see it, it seems obvious and you cannot un-see it. But seeing is difficult until you know where to look.

Here is a somewhat humorous hypothesis for you to consider: the primary difference between myself and actual for-real AI experts is that I did it on bare metal in octal for 20+ years.

But if that hypothesis happens to be true, it has an implication: 20+ years of close observation, preceded by 40+ *prior* years of accumulated industry tradecraft, shows how to use AI in revolutionary ways to solve problems not considered solvable today. At Cray Research, "it cannot be done" was never a barrier. It was the necessary starting point.

Tradecraft Transmission

Tacit knowledge has traditionally been transmitted from person to person through demonstration, collaboration, and mentorship. But the people with the knowledge, at least in non-classified environments, are retiring out of the workforce.

Our knowledge and engineering tradition *can* be passed in written form, with the mentor/author not present. The next chapter shows how.

Chapter 2. Apprenticeship Transmission

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Code Generation as Example

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Visibility Lost

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Transmission Method

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Book Design

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Conditions for Formation of Thinking

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Example Usage

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Chapter 3. Teaching and Creating Mastery

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Road Versus Map

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Training Data Cutoff Date

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Waypoint Details

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Multiple Information Layers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Parallel and Equivalent Routes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Shifted Perspective

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Same Pattern Different Context

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Classroom Practice

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Mental Models

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Visibly Modeling Expert Thinking

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Thinking About Thinking

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Recursion as Technique

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Classroom Question Answering

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Lived Experience

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Three-Step Process

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Mental Models Enable Analogies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Chapter 4. Transcendent Patterns: Teaching the Process of High-Tech Mastery in Student-Accessible Fashion

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Definition

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Transcendent Pattern

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Invariant

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Fun and Superpower as Motivation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Tools of Transmission

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Three-Step Process

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Transcendent Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Invariant

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Integrating the Skill

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Practicing the Insight

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Design Pattern: Struts and Braces

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

World War II Biplane

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

World War I Biplane

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Design Pattern: Cantilever

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Swing Bridge

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Transcendent Insight

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Teaching This Skill

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Examples at Hand

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Hypothesis as Transcendent Pattern

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Cognitive Design of This Chapter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Chapter 5. If You Want More

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

The Barrier

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Capabilities

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Solutions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Cognitive Design of This Book

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Separate Concept From Implementation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Parallel Lineage

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Novelty

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Content as Method

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Creating Agency

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Declassified Documents and Road Map

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Appendix A. Analytical Machine Employment (1952)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Purpose of Analytical Machines

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Machines according to Potentiality

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Employment Related to Potentiality

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Present Employment Versus Reserve Capacity

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Employment of the Revolutionizers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Employment of the Labor-Savers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

IBM

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

ROBIN

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

GOLDBERG

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Miscellaneous Small Operations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Discussion

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

IBM's Labor-Saving Employment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

The ROBIN Job

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

ROBIN Utilization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

ROBIN Preparation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

GOLDBERG

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Miscellaneous Small Operations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Appendix B. Cryptanalytic Machines in NSA (May 1953)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Dr. Howard Campaigne

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Bombes, Scritchers, and GOLDBERG

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/tradecraft.

Appendix C. HPC Tradecraft Road Map

What is "HPC tradecraft" as I know it? This is high-performance computing (HPC) arising from signals intelligence (code breaking) needs. My particular lineage begins with the "Tunny" code breaking project at Bletchley Park during World War II. That produced such an odd-looking contraption that the "Wrens" operating it named it the Heath Robinson. W. Heath Robinson was a cartoonist known for portraying fantastically complex machines designed to accomplish simple tasks.

When U.S. Navy veterans built something similar for a similar purpose, they named it GOLDBERG in honor of cartoonist Rube Goldberg's fantastically complex machines, explicitly acknowledging the Heath Robinson tradecraft origins. One of the founders of Engineering Research Associates (ERA) was Bill Norris. ERA built GOLDBERG, and also hired Seymour Cray. Norris later founded Control Data Corporation (CDC) and Cray soon followed as their youngest employee. Cray later founded Cray Research, which I joined early 1980.

Thus the high-performance tradecraft I learned within Cray Research already had a 40-year history. Perhaps because so much was classified Top Secret, we never wrote down *how* we did things. Institutional knowledge was passed from person to person as mentorship, collaboration, and informal apprenticeship.

The 1995 Barrier

Circa 1995, something shifted. We hid the messy details behind abstractions, allowing for far more complex software development. We no longer needed to think of systems as a whole, and how things performed on bare metal.

For high-performance computing, this presents a problem. HPC tradecraft remains constraint-based design, but we can no longer see the constraints informing the design. This became a niche topic rarely covered in school (or the workplace).

Meanwhile, the pre-1995 tacit knowledge is retiring out of the workforce. My role as Custodian (the Russian term is more specific, хранитель) is to share this knowledge with you as a living practice.

Two Series

I created two book series to teach HPC tradecraft.

- "HPC Tradecraft Apprenticeship", three books, is the institutional tradecraft as I learned and practiced, and still practice. Tradecraft it useless unless replicable. The first book teaches my replication method; the second teaches *how* we accomplished what we did; the third teaches systems thinking, with close observation and characterization of HPC systems under load, using modern AI as the example. The three books are transmission protocol; HPC design; HPC operations.
- "The HPC Tradecraft Master Practitioner", three books, is the author letting loose and having fun, expressing his mastery of HPC tradecraft. High-Stakes Ethics prepares for the insurmountable power imbalance of a new graduate in the amoral corporate environment. Unexpected Histories demonstrates debugging skills applied outside computer science, showing their value and timeless nature. Constraint-Based Design demonstrates 1986 tradecraft directly applies to modern AI transformer design and usage.

Book 1. HPC Tradecraft for Computer Scientists: What We Stopped Teaching

I iteratively created book 1 as I circled what I actually have to share, why it is important, and how to make it accessible to the right persons.

- Show my teaching method of formation text that performs what it teaches, which I frame as the manner of replication.
- Act as filter and qualification, so that readers have the necessary information before buying/committing to the two large flagship books.
- State *Nobody but Us* as prerequisite to *Wizard's Lens*, which readers might perceive as problematic if they came for AI.

The sequence itself is formation. This was the teaching order within Cray Research.

Book 2. Nobody but Us: A History of Cray Research and the Building of the World's Fastest Supercomputer

This book is by far the largest and highest writing quality. However, persona walk-throughs consistently report the quality as uneven. This is because the book attempts to overcome the Edwin Abbott Flatland paradox. "Look here" in the midst of formation text appears to be uneven quality requiring an editor, and a human might see it the same way, but "look here" is pointing to something most readers do not have the vocabulary to see on their own.

The book serves a crucial purpose of credentialing the author (via demonstration not assertion). Why? Because the next book makes apparently-outrageous claims which are not proven until reading through to the final chapter. The claims are based in "Nobody but Us" tradecraft.

What Cray Research accomplished is well documented. This book teaches *how* we did it. But there is another aspect, which is the next book.

Book 3. The Wizard's Lens: Learn to Think Like AI

Think Jay Forrester and Donella Meadows and Eli Goldratt and Robert Gagné and Anders Ericsson, all of whom are referenced in the book. This is the craft of closely observing and characterizing HPC systems, using AI as the working example. This is the Cray Research attitude of treating barriers as opportunities. Barriers point to the point of maximum leverage. Meadows lists points of leverage. For my purposes, I collapse her list to Liebig's Law of the Minimum (plant growth is bounded by the relatively scarcest nutrient). This principle matches Goldratt's *The Goal*.

Rather than the skills in the prior book, this teaches shifting perspective, systems as a whole, interactions between systems at the boundary, and the attitude of not being distracted by the fact that it has never been done before.

This ends the primary series, the HPC Tradecraft Apprenticeship.

Book 4. High-Stakes Ethics

The book is a work in progress. My premise is that we do not give computer scientists, particularly recent college graduates, the skills to survive the corporate power imbalance. I wrote two essays regarding that power imbalance. They are the raw material from which I plan to write the book, but the essays themselves have urgent importance, thus publishing as a work in progress.

Book 5. Unexpected Histories: Spotting Patterns and Making Connections That Others Miss

This book consists of 15 essays/chapters, 49,000 words, demonstrating tradecraft in primary source analysis and constructing interesting narratives by "connecting the dots" that others miss. While nothing in here is technically difficult, some seem to have this aptitude and some do not. Book 2 has a chapter on what William Friedman has to say about developing intuition. This book is practice in developing that premise.

Tradecraft demonstrated across the centuries and outside computing demonstrates invariants, skills non-specific to a given technology, while demonstrating Piotr Galperin's final stage of formation.

Book 6. Constraint-Based Design: A Gateway to AI

In 1986, I wrote a bare metal device driver inside the Cray I/O Subsystem as a joke and to demonstrate the author's capability. It is a text adventure game, Swiss Adventure, inspired by Colossal Cave Adventure. What is useful is its characteristics.

The game itself forces the user (adventurer) into an operational apprenticeship, working with incomplete information in an unfamiliar context. I created a PHP/Laravel/React.js version that is live on my website, stateless, requiring no login or registration or data collection. The live version is the game itself presented as typing at the operator console, with side displays (essentially event traces) showing flow through the assembly language overlays, and flow through the LLM attention mechanism patterns.

Thus it is an executable companion to the book. The 1986 source code and the PHP engine are in a public GitHub repository, with the most recent timestamp in the repository as May 2019, which is well before GPT and Claude became well known publicly.

- The live website is here: Swiss Adventure
- The assembler and PHP source code are here: SwissAdventure

There are several aspects of interest.

UPDATE

The source code is in UPDATE format. Secondary sources (wikipedia) indicate CDC UPDATE is one of the earliest known source code management systems, from late 1950s or early 1960s, punch card based. I know from personal (as primary source) knowledge that the first Cray Research software person was hired directly from CDC. I know from internal statements that CDC's UPDATE was ported to Cray Research. While onsite at Boeing Computer Services 1980-1982, the front end systems were CDC Kronos. I used CDC UPDATE and Cray Research Update side by side. Thus I am clear on the provenance.

Relatively few extant examples of live code in UPDATE format exist. This one is publicly available on GitHub.

APML

When the CRAY-1/S was announced with a new I/O Subsystem (IOS) in addition to the mainframe CPU, the IOS had been developed as the "A" Processor. Extant documentation as of 1980 called it that. The cpu assembler was CAL, Cray Assembly Language. The IOS assembler was called APML, "A" Processor Macro Language.

The 1986 source code is a live, complete, package, and likely the only substantial example of IOS programming written by an actual Cray Research IOS programmer. It was written (as a student) during a class *teaching* IOS programming, but I was later part of the IOS group for 10 years, so can act as primary source asserting it is authentic and representative.

"Live" might be misleading in that no system still exists to run it. But the precise port to PHP *is* live and online.

Adventurer Experience

The adventurer experience has characteristics similar to many realms of operational training. I intentionally wrote it inspired by Colossal Cave Adventure and Zork. It might be worth noting that the authors of those two programs worked in operational environments. Thus I was not inventing the formation aspect. That was already a characteristic of text adventure games.

Constraint-Based Design

The source code has many levels and aspects of constraint-based design, and it is possible to teach the design without dwelling on assembler syntax. I implemented the training data (all navigation and text content) as a domain-specific language using the assembler macro and micro facility. I had a deep enough understanding (including making mods to the CAL assembler source) to do this. Adding game features did not require executable code changes.

Here is the odd part: The source code architecture is isomorphic to 2017 "attention is all you need" architecture function. Similar constraints shape similar solutions (Genrich Altshuller's TRIZ principle). My operating environment was quite similar to a hot LLM token context under load. Attention heads, weighting, pulling from external source then deallocating... quite a few features are present.

A working bare-metal example of constraint-based design can be useful. But a 1980s demonstration that HPC methods are relevant to modern AI has interesting implications, which is why I decided to include this as a book.