Course Overview

1. **Choosing Research Projects**

   Research Life Cycle

2. **Methods and Supporting Skills**

   Practice with methods not covered in other classes.

3. **Professional Career Issues**

   Proposals and Presentations

Units of Study

Each area of computer science chooses different units of study.

- In algorithms: algorithms
- In AI: methods, techniques, algorithms
- In Languages: languages, language components or features
- In Architecture: instruction sets, memory hierarchies, architectures
- In Theory: models, theorems, proof techniques
- In Systems: systems, components of systems, system architectures

Units Have Hierarchical Structure

- architecture
    - instruction sets
        - instructions for procedure call/return
        - instructions for branching
    - memory hierarchies
        - support for virtual memory
        - study of caches
        - study of cache coherency
    - floating point units
        - floating point representation
        - arithmetic algorithms
        - implementation strategies
        - pipeline design

Unit Diversity

Units vary by their size, potential variety, cost of evaluation, and so on.

Size:

Small units: algorithms, language features, architectural components

Medium-size units: instruction sets, proof techniques

Large units: languages, architectures, machines

Potential Variety:

Low variety: sorting algorithms, cache designs

High variety: AI algorithms, languages, architectures

Cost of Evaluation:

Low cost: algorithms, AI methods

Medium cost: theorems, components of architectures

High cost: languages, architectures, ideal models

Generally, smaller units exhibit less variety and lower cost of evaluation, so they are easier to do research on.

Research Life Cycle

- **Definition.** Exploratory research defines a new problem, new constraints, new opportunity, or a new approach.
- **Initial Solutions.** Initial algorithms, designs, theorems, programs are developed.
- **Evaluation of Initial Solutions.** Initial solutions are evaluated and refined in isolation.
- **Comparison of Solutions.** Solutions are compared to one another and also to ideal solutions.
- **Space of Possible Solutions.** Theorems are proved about the limits on any solutions. Existing solutions are placed in a common framework to determine whether all possible solutions have been found.
- **Technology Transfer.** Best approaches are transferred to users.

Not all of these phases are seen in all areas. For units with high cost of evaluation, only relatively weak methods can be applied to evaluate initial solutions and compare solutions.

For units with high variety, it is difficult to understand the space of all possible solutions.

Research Results

- a definition of a problem or task
- a unit for solving a problem, performing a task
- identification of factors influencing the cost, effectiveness, or applicability of a unit (perhaps with some idea of the relative importance of the factors)
- development of an ideal model
- a finished unit that can be distributed to users
- measurement of some properties of a unit: run time, chip area, representation requirements, reliability, useability, etc.
- identification of problems and shortcomings in a unit.
- a demonstration that one unit is better than another.
- a definition and demonstration of a tradeoff
- analysis of a tradeoff showing how different points on the curve can be obtained and selected.
- a generative (explanatory) theory for some set of units

Research Methods

- writing programs
- writing systems
- developing architectures
- developing content architectures (ontologies, knowledge bases, class libraries, graphics toolboxes, etc.)
- measuring properties of units
- finding and proving theorems
- analyzing and consolidating previous research
- interviewing experts, customers
- performing psychological experiments, surveys, observations

Research Methods (2)

- building hardware
- reading literature
- importing techniques and results from other fields
- measuring and predicting constraints on future units (eg., VLSI technology, government regulation, user expectations and requirements)
- writing papers, monographs, and textbooks

Research Project Phases

An individual research project (such as a Ph.D. dissertation) follows a lifecycle related to the research life cycle:

- Choose research question/problem/tradeoff.
- Determine current state of knowledge (lifecycle phase)
- Apply appropriate methods to produce research results.
- Write up research results

Research is not complete until it is written up!

Supporting Skills

Different research methods require different supporting skills.

- programming (and hardware design)
- organization
- mathematics
- psychological techniques: protocol analysis, experimental manipulations, survey methods
- statistics
- writing proposals
- writing papers
- critiquing papers
- designing experiments
- giving talks

You study some of these methods in other courses. This course will attempt to cover the remaining ones.

Lifecycle Readings

You must choose one of three areas

- concept learning
- RISC computer architecture
- constraint logic programming

and read a series of papers to understand the research lifecycle. You will write a brief report describing each paper.

Paper Critique Outline

- **What is the research goal?**
  What question(s) is the author trying to answer?
- **What methods are being applied?**
  What methods is the author applying to answer the question.
- **What are the research results?**
  A paper can contain many different kinds of results.
- **What claims are made in the paper?**
  For theoretical papers, what results are proved?
- **How are these claims supported?**
- **What reasonable claims and results are missing from the paper?**
- **What would be reasonable next steps for the research?**

Phase–Method Matrix

	Design units	Implement units	Run units	Find & prove theorems	Analyze and consolidate	Study users	Import techniques	Read literature	Write papers
Definition	X					X		X	X
Create units	X	X				X	X	X	X
Evaluate units		X	X			X		X	
Compare units	X	X	X	X		X		X	
Space of units									
Ideal model				X	X			X	X
Tradeoff					X			X	X
Explanatory Theory					X			X	X

Evaluation Criteria

Experimental research requires measurable evaluation criteria.

- **CPU Time**
- **Cycles per instruction**
- **Percentage of correct classifications**
- **Number of serious interface flaws identified**

Ideally, these criteria will be

- **Easy to measure**
- **Reliable** (i.e., replicatable)
- **Valid** (i.e., the measure the right thing)
- **Applicable Early in the Design Process**
- **Convincing**

Evaluation Criteria Must Implement Goals

It is very important to consider the relationship between measurable criteria and the underlying goals of the project.

Can a unit score well on the criteria and yet still fail to achieve the underlying goals?

Three Examples

Example 1: CISC

Experimental evaluations of architectures focused on

- **Code size** (in bytes)
  Entropy of Op-code field.
- **Orthogonality** (all modes can be combined)

Efficiency of execution of the resulting programs was not being directly considered.

Evaluation criteria are "proxies" for the true goals:

- **code size**.
  True goal: efficiency (reduced instruction fetch, page faults)
  True goal: cost-effectiveness (reduces memory cost)
- **orthogonality**
  True goal: ease of assembly-language programming

RISC showed that the connection between the criteria and the underlying goals was no longer strong.

Example 2: MYCIN

MYCIN: An expert system for diagnosing bacterial infections in the blood.

Study 1 Evaluation criterion:

- **Expert Ratings of Program Traces**
  decide whether patient needed treatment?
  determine significance of isolated organisms?
  ability to select therapy?
  overall consultative performance?

This had several problems:

- **Assumed that Experts were ideal diagnosticians**
- **Experts knew they were evaluating a computer program** (and may have been biased against it)
- **Too much expert time required**
- **All experts were from Stanford Hospital**

MYCIN Study 2

Study 2 Evaluation Criterion:

- **Expert Ratings of Treatment Plan**

Experts studied the charts and studied MYCIN's recommendations and rated them using a multiple-choice table.
Experts were from several hospitals.

- **Assumed that Experts were ideal diagnosticians**
- **Experts still knew they were evaluating a computer program**
- **Too much expert time still required**

MYCIN Study 3

Study 3 Evaluation Criterion:

- **Expert Ratings of MYCIN Treatment Plan and several other human-supplied treatment plans**

Raters didn't know whether they were evaluating a computer program.

Raters were evaluating novices, intermediates, and experts as well as the program.

Results

Prescriber	% acceptable by an evaluator (n=80)	% acceptable by majority of evaluators (n=10)	Number of cases where therapy failed to cover all
MYCIN	65.0	70.0	0
Faculty-1	62.5	50.0	1
Faculty-2	60.0	50.0	1
Fellow	60.0	50.0	1
Faculty-3	57.5	40.0	0
Actual Therapy	57.5	70.0	0
Faculty-4	55.0	50.0	0
Resident	45.0	30.0	1
Faculty-5	42.5	30.0	0
Student	30.0	10.0	3

- **Experts Do Not Agree**
- **The Experiment is Valid**

  People with more experience rate better than people with less experience.
- **MYCIN is doing well**

Lessons Learned

- **Don't assume experts are perfect**
- **Find out how humans are evaluated**

  You may need to develop your own methods for evaluating people as well as machines.
- **Control for potential biases**

  Controlled for human vs. computer; Stanford vs. other institutions; expert vs. novice
- **Don't expect superhuman performance**

  In MYCIN, this would have involved evaluating the system according to what the right answer eventually turned out to be (e.g., by the eventual results of a blood test). However, diagnosis and treatment is performed under uncertainty—human experts are not evaluated according to the "right answer" either.

Example 3: User Interface Design

How should a user interface design be evaluated?

What are the goals of evaluation?

- **Debug the Design** Provide early feedback to designers about potential bugs.
- **Choose Among Alternative Designs**
- **Compare Design Tools**

Four UI Evaluation Methods

- **Usability Testing:** Construct a set of tasks to perform using the interface. Have human subjects carry out these tasks. Measure their speed and accuracy and the number of problems they encounter.
- **Heuristic Evaluation:** Have one or more UI experts evaluate the design to identify problems.
- **Cognitive Walkthrough:** Designers walk through a task with the interface, answering a set of questions for each action.
- **UI Design Rules:** Evaluate the UI against a series of design rules.

A Comparative Study

	Usability	Heuristic	Walkthrough	Design Rules
via technique:	30	105	30	13
side effect:	1	—	5	8
prior experience:	0	—	0	12
total:	31	105	35	33

Curiously the "gold standard" (usability test) did not identify as many problems as the Heuristic Evaluation.

Evaluation Criteria in Research

- **If you don't know how to evaluate what you have done, you won't know if you have succeeded.** And you won't be able to convince other people either.
- **Defining evaluation criteria can be an important research activity itself.**
  In the early phases, it is part of the problem definition.
  In later phases, new better proxy criteria are often identified.
- **Early feedback from criteria can guide research.**