Research in Experimental Computer Science and Engineering

Bengt Ahlgren
SICS
Background

“Academic Careers for Experimental Computer Scientists and Engineers”

- Report from the Computer Science and Telecommunications board, National Research Council on request from NSF
- Explains need for ECSE
- Focuses on the special challenges faced by faculty who wish to be successful experimentalists
- Deals primarily with career track of regular faculty members (“tenure track”)

Report outline

- What is Experimental Computer Science and Engineering?
- An academic career in ECSE
- Educational dimensions of academic ECSE
- Evaluating research in ECSE
- A positive environment for academic ECSE
- Special needs and concerns of non-doctorate-granting and less recognised institutions
“Computing the Future”

CSTB report that characterises CS&E as a field:

“Computer scientists and engineers focus on information, on the ways of representing and processing information, and on the machines and systems that perform these tasks.”

Key intellectual themes:

- algorithmic thinking,
- information representation, and
- computer programs.
“Computing the Future” (cont.)

Five subdisciplines:

- systems and architectures;
- programming languages, compilers and software engineering;
- artificial intelligence;
- computer graphics and user interfaces;
- algorithms and computational complexity.

Experimentation is an important aspect of many of these.
“Computing as a discipline”

“The Denning report” – enumerates subareas:

- Algorithms and data structures
- Programming languages
- Computer architecture
- Numeric and symbolic computation
- Operating systems
- Software engineering
- Databases and information retrieval
- Artificial intelligence and robotics
- Human-computer interaction
“Computing as a discipline” (cont.)

Classifies each subarea via three basic processes:

- theory,
- abstraction, and
- design

that are used by the subareas to accomplish their goals.

*However*: ignores the role of experimentation.
Defining characteristics of ECSE

- It is a synthetic discipline
- It focuses primarily on *artifacts*
- The artifacts of ECSE are extraordinarily complex
- It is sensitive to technological developments
- Computing artifacts are universal
- It is not strongly coupled to theoretical computer science
It is a synthetic discipline

The phenomena studied by most practitioners have been created by a person rather than being “given” by nature.

This characteristic is shared with other branches of CS&E.

Complication: hard to assess the contribution embodied in an artifact because the synthetic property under-constrains them.
Focuses primarily on artifacts

- The artifact can be the subject of study, the apparatus with which to conduct the study, or both

- Artifacts often embody a substantial portion of the intellectual contribution of experimental research

- The creation of artifacts represents a significant intellectual effort

- Artifacts serves three primary purposes/roles:
  - Proof of performance
  - Proof of concept
  - Proof of existence
Artifacts

Not only electronic hardware or software systems. Also including:

- graphic images and animations,
- robots,
- certain hard-to-construct data files including multi-processor execution traces,
- test benchmark suites and
- structural descriptions (such as the Utah Tea Pot)
Proof of performance

An artifact acting in the proof-of-performance role provides an apparatus for direct measurement and experimentation.

- The artifact exists or can be constructed
- The results produced are usually quantitative
- Perhaps the most typical artifact of ECSE research

Example: peephole code optimizer.
Proof of concept

An artifact acting in the proof-of-concept role demonstrates by its behaviour that a complex assembly of components can accomplish a particular set of activities, behaviour that could not be argued simply by logical reasoning or abstract argument from first principles.

The working system, the artifact, is a witness “proving” that the concepts in at least one configuration are correct.

Example: experimental computers implementing new concepts for hiding memory latency.
Proof of existence

An artifact playing the proof-of-existence role conveys the essence of an entirely new phenomenon.

Example: computer mouse.

Least common of the three roles.
The artifacts are extraordinarily complex

- Both construction and dynamic behaviour are complex
- Creating and understanding artifacts can require substantial intellectual effort
- Complexity takes several forms, including a large number of components and high component specialization
It is sensitive to technological developments

- Intimate relationship with technology
- Technology in which an artifact is implemented is not an incidental aspect
- The use of cutting-edge technology can be risky
- Not using cutting-edge technology can be risky
- Dependence on technology (HW and SW) developed by others
Computing artifacts are universal

Computers are malleable and versatile – they are universal, which means that within broad limits, whatever one machine can do, all machines can do.

Serious complication: there is no a priori limit on the functionality of computers, which leads to ever-expanding expectations for the capability of artifacts.

Expectations for increased functionality can affect other systems that serve similar purposes.
It is not strongly coupled to theoretical computer science

- Unlike, e.g., physics, an “experiment” in ECSE generally does not verify a prediction from theoretical CS, or rely on a model developed theoretically.

- Reason: complexity of most real problems precludes the direct application of analysis.

- A problem can be made theoretically tractable only by abstracting so extensively that the problem that emerges may not capture the essence of the original problem.
It is not strongly coupled to theoretical computer science (2)

However:

- Good experimental work is grounded in testable models and hypotheses

There are also exceptions:

- Language theory underpins parsing in compilers
- Complexity theory underpins data encryption
A succinct definition

“ECSE involves the creation of, or the experimentation with or on, computational artifacts.”
What makes ECSE research?

- Synthetic -> straightforward to create new artifacts
- Not automatically an intellectual contribution
- Must be shown to be better than some alternative
- Proof of performance: objective measure
- Proof of concept or proof of existence: “better” may be a subjective human judgement
What makes ECSE research? (2)

“What making an artifact is research when it contributes directly or indirectly to our understanding of computing.”

Implies two specific requirements:

- The artifact must embody some computational phenomenon in a manner that reveals new information, and must be constructed in a way that conveys the information reliably.

- The new information is extracted from the artifact and conveyed in a suitable medium and scholarly manner, i.e., the implementor must teach others.
Evaluating research in ECSE

Main problem: fewer journal publications compared to theoreticians.

Conference publications are much more important because of the rapid technology advancement.

Other forms of dissemination are very important, specifically the distribution of artifacts.