

Building an Internet of Things Platform

Sabbatical Leave Research
Presentation (2024 - 2025)

Table of Contents

- Research Overview
- Research Details
- Impacts on Teaching
- Future Directions

Research Overview

The Domain

- The Internet of Things (IoT)
 - Understand IoT Programming Model
 - Application Development Platform
 - Building Tools for IoT Development

The Problem

- Compared to conventional desktop, web, or mobile, IoT application development environments are *heterogeneous*.
- People need to understand software, hardware, and communication factors to build an IoT system.

The (Proposed) Solution

- We need a new **IoT programming model** based on the understanding of IoT development.
- Based on the programming model, a new kind of **IoT development platform** is proposed and implemented.

Research Goals

1. **Define** IoT Platform
2. IoT Platform Architecture **Design**
3. **Develop** the Platform

If Possible (Not in the plan):

4. **Deploy** the Platform to make the IoT Portal.

Goal 1: Define IoT Platform

Approach:

- Analysis of existing IoT devices and development tools
- Definition of **programming abstractions** for IoT systems

Goal 2: IoT Programming Framework Design

Approach:

- Use the Ontology as the tool for search & build.
- Design of **scalable** platform architecture from the *Define*.

Goal 3: Develop the Platform

Approach:

- Feasibility validation through **working** implementations based on the *Design*.
- User-centered feature development and testing

Research Results

- Goal 1 (**Define**): Ontology-based IoT programming model.
- Goal 2 (**Design**): IoT query language and database tools for the IoT model.
- Goal 3 (**Develop**): An IoT platform using Obsidian as the DBMS System.

- (Not in the proposal) **Deploy**: Share the platform with GitHub actions with the Obsidian Database.

Research Details

Define: IoT Programming Ontology Model

Why is an ontology needed?

- Ontology gives IoT a shared **vocabulary**.
- It abstracts device details with **standard** terms.
- We can use it to support **reasoning**.

How Ontology is used in Programming Model

- Using an Ontology, the entities in IoT can be explicitly defined.
 - Microcontroller (M)
 - Sensor (S)
 - Actuator (A)
 - Communication (C)

- Ontology defines relations among entities.
 - e.g., Microcontroller 1 has a sound sensor.
 - Microcontroller 2 lacks communication, so add a board.

Insights Earned

- Ontology relationships fit naturally in a Graph DB; reasoning is just graph search.
- IoT programming model can take advantage of this model:
 - **Entities** as objects
 - **Relationships** declared

Design: IoT Query Language and Database

- IoT Ontology defines entities and relationships.
- Queries use this notation to retrieve related info.

Model & Relationship Definition

- The relationships among the entities are defined and can be reasoned.
 - $m1 \text{ (MSP430)} \in M$
 - $s1 \text{ (Gas Sensor)} \in \text{Sensor}$
 - $m1 \diamond \rightarrow s1 : \text{MSP430 has Gas Sensor}$

Query

- $m? \diamond \rightarrow s1$ (What microcontroller has a gas sensor?)
 - $m1$ (MSP430)
- $m? \diamond \rightarrow s?$ (Show me any microcontrollers that own any sensors.)
 - $m1 \diamond \rightarrow s1$

Other Utilities

- `information(m1)` (show me the information about MSP430)
- `example(m1 $\diamond \rightarrow$ s1)` (show me example that MSP430 uses Gas Sensor)

Database

Findings:

- SQL works, but tables aren't ideal for this model.
- A document DB (NoSQL) fits better: Markdown + YAML + JSON.
- We can use the existing Query Language (SELECT, FROM, and WHERE) instead of detailed coding

IoT Platform based on Obsidian and GitHub

Transitions:

1. Ver 1 - PyQt6 Library: Python
2. Ver 2 - VSCode (Electron Platform):
JavaScript
3. Final - Obsidian (Electron
Platform): JavaScript

Ver 1 - PyQt6 + Python

- **Rapid Development:** Python's simplicity speeds coding
- **Flexibility:** Combines Python libraries with Qt power

However

- Building the IDE and DBMS alone was **too much work.**

Ver 2 - VSCode Extension

- Using **VSCode** instead of building an IDE is the next choice.
- Since VSCode is in **JavaScript**, extensions are easy to develop.

However

- My IoT platform model needs a DB for **entity definition + search**.
- **VSCode** lacks DB/search per se.

The Choice - Obsidian PKM System

- I found **Obsidian** fits perfectly for my IoT platform.
- It's a **Knowledge Management tool** built on Electron (like VSCode).



Extending Obsidian

- Extend Obsidian with JavaScript.
- I could add any features necessary using the defined ontology.
 - \in corresponds to a function `displayIn`.

Example

- $m? \in \text{"Sensor"}$ (Find a controller that has a sensor)

This Ontological representation can be translated into JavaScript.

```
``datajs
let m = "MicroController?"
let s = "Sensor"
const {MocGen} = await cJS()
MocGen.displayIn(dv, m, s);
``
```

Microchip is in the IN property of these pages (Total 1)			
L(1)	mark	link	path
M		Microchip Product (Microchip Product)	Source/M/Microchip/Microchip Product

AI Coding Assistance

- Wrote about 3K LoC in JavaScript in 2 months (expected 20K - 50K when submitted proposal)
- Discovered Vibe/Low-code tools
- Tried Claude, Perplexity, Windsurf, Cursor -> coding burden disappeared

- Coding **speed & quality** improved dramatically
- Could finish most coding with just a few clicks - only ideas were needed
- Like hiring a genius programmer for \$20/month
- The more I learned about GenAI, the better results I got

Bonus: GitHub as the Deployment Tool

Finishing the coding quickly allowed me to focus on the deployment strategy.

- Share and distribute ontology-based knowledge database via GitHub
- Use GitHub Actions for automation, easy distribution, and community contributions to the IoT platform
- AI taught me how to use it.
- AI made working code and accompanying tests, and Docker examples.

Bonus: Utilities

I wish I had made some utilities while building the platform.

- e.g., managing multiple repositories across multiple GitHub IDs
- Estimated at least 1–2 weeks for design, coding, and testing.

- I gave up even though I know how to make one; I didn't have time.
- I just asked, and AI built it in **1 hour** (a few iterations).
- Created more custom utilities for my work with AI-assisted/Vibe coding: enhanced productivity and automated work process
- The era of manually writing code seems to be ending

Impacts on Teaching

New Course Development

CSC 494: AI-Driven IoT System Development (Sprint 2026)

- Students build IoT projects with ESP32/Arduino, BLE/I2C, and sensors, connecting to mobile and server via Docker.

Generative AI supports learning, building, and debugging.

- Students will use "C" to explore IoT devices with the help of AI.
- Students will use the PKM system to learn IoT topics and make systems with AI.
- Students will use tools/techniques/rules to use AI effectively.

Future Directions

New Research Direction

- Adding IoT reasoning to the Platform: LLM + Expert System (DB)
- Supporting Design diagrams (UML-like) for problem focus
- IoT-dedicated LLM (ChatGPT model) for development

New Teaching Directions

- Update my courses, adapting to changes.
- Make more courses using AI as a tool for learning ideas and building software in different domains.

Any Questions?

Thanks for your attention

I could complete my planned research **earlier** and in **better quality** than I expected, thanks to the AI assistant. I am using the tools, PKM, and utilities that I created (with AI) every day.

So, I started my 2nd research project.

AI for Success in ASE Education

How to make students 10x software
engineers

Five Topics to Share

1. Enlightenment Moment
2. Go Industry → SWE Industry
3. SWE Industry: Coding Is Over
4. New Opportunities
5. My Approaches/Efforts

Enlightenment Moment

To design an IoT database query system, I had to solve this coding problem.

Q: Write a JavaScript function that parses the input and executes `f()` depending on the order.
For example, `!a & (b || c)`, it executes `!((f(b) || f(c)) & f(a))` automatically.

My Implementation

- Lexer/Parser/Executer (+ 100 LoC)

```
function evaluateBooleanExpression(expression, f) {  
  // Tokenizer  
  function tokenize(expr) {  
    const tokens = [];  
    let i = 0;  
  
    while (i < expr.length) {  
      const char = expr[i];  
  
      // Skip whitespace  
      if (/\\s/.test(char)) {  
        i++;  
        continue;  
      }  
  
      // Operators and parentheses  
      if ('!&|()'.includes(char)) { ...
```

LLM Answer in 3 secs (3 LoC)

It even teaches me how to do it step by step.

```
function evaluateBoolExpr(expr, f) {  
  // Step 1: Replace variables with f() calls  
  const withF = expr.replace(/\b([a-z_$][\w$]*)\b/gi, 'f("$1")');  
  
  // Step 2: Convert to valid JS operators  
  const jsExpr = withF.replace(/&/g, '&&').replace(/\|/g, '||');  
  
  // Step 3: Create a safe evaluation function  
  return new Function('f', `return ${jsExpr}`)(f);  
}
```


Same results:

- $< 4\%$ of LoC, within 3 sec, and a new perspective
- Easy to understand, use, and maintain
- It can generate the Lexer/Parser version too if I ask!

```
const result = evaluateBoolExpr('!a & (b || c)', (varName) => {  
  // Your custom logic for each variable  
  console.log(`Evaluating ${varName}`);  
  return true; // Return actual boolean value  
});
```

- I don't know if the LLM found the most straightforward solution online or reasoned about the logic.
- It doesn't matter to me, as it not only **solves** my coding problems, but also **directs me** the correct way; even **showed** how to do it.
- In a way, this is both discouraging and eye-opening.

- If I keep teaching students the same way I always have, I might run into problems.
- But maybe, with the right approach, I can help ASE students become 10x better programmers.

So, I had to start my second research unexpectedly.

2. Go Industry → SWE Industry

- After the AlphaGo match, Lee retired from Go three years later, knowing the human race cannot match AI forever.





- The power of knowledge/training + reasoning = Super Go Player (Go god)

- He recently published a book to share his insights and experiences with AI.
- We can learn from him to understand what will happen to SWE.



What has happened in Go

1. Humans developed Go joseki (patterns, 정석/定石) over thousands of years
2. AI discovered these patterns—and revolutionary new ones—in days
3. We compressed centuries of accumulated wisdom into machine learning

Go schools devastated, Many Competition Prices gone

1. Everybody knows that the Go champion is AI, so Go schools in Korea are devastated.
2. Even pro Go players learn from AI.
3. We still have Go competitions, but prices go down.
4. People play Go in the way that AI teaches, not in the traditional

The Same Thing is Happening in SWE

- AI codes better than **99.9%** of humans
- Soon → **100%** (maybe already now)

- At the AtCoder World Tour Finals 2025 in Tokyo: **Przemysław "Psyho" Dębiak** (Poland) won **1st place**, **OpenAI's model** ("OpenAIAHC") came in **2nd**, narrowly behind at ~9.5% gap.

We are forced to think about the meanings of them

- What's the meaning of the Go championship when all players learn Go from an AI (Go god)?
- Everybody knows the way to be the Go champion (if they can cheat, but not be detected cheating).

- What's the meaning of coding competition, i.e., ICPC, when AI can solve any of the coding/programming problems?
- Anyone can hire Coding god for \$20/month.

Just a New Tool, or More than that?

- We should find a new meaning of competition.
- I believe we just found a new tool to revolutionize the human race as a whole
- Or, is it much more than that?

Google CEO Sundar Pichai: "AI more profound than the discovery of fire or electricity"

- We cannot go back; Maybe we are in an "Adapt" or "Perish" situation.
- Do we need another Luddite movement (like 300 years ago)?

We need a policy that everybody can agree upon

- Can we say to students, "Don't use AI because you need to learn code the way I did?"
- Can we prevent students from using AI when they can get all the answers with one click?
- We need to guide students.

SWE Industry: Coding Is Over

Meta CTO: AI in engineering =
internet-level shift

→ Developers gain huge leverage if
they **adapt**

This is the point we should focus on:
the industry doesn't need coders
anymore; it requires much more than
that from the beginning.

Dario Amodei, CEO of Anthropic, predicts that within **3 to 6 months**, AI could write all code within a year.

- I believe it is more than 99% now.
- From my experience, managers may not allow software engineers to code, but force them to **solve really important SWE problems**.

Recent research from Clutch reveals that **over half of developers think AI LLMs can code better than most people.**

Once anyone uses it, no one can deny it; LLM knows everything about coding like a genius programmer or coding god.

92% of U.S. devs use Copilot, ChatGPT, etc: SWE-bench scores backing up their effectiveness.

- More and more companies allow LLM usage for their job interview.
- Managers already know who the coders or problem solvers are, so the firing or switching job functions.

The 2025 SWE-bench results show a giant leap: **AI's successful coding problem-solving jumped from just 4.4% in 2023 to 69.1% in 2025**, dramatic improvements in AI coding capabilities and realism.

This matches the moment when I first became aware of the power of Low/Vibe coding, which I had ignored until 2024.

We are familiar with changes

120 Years ago

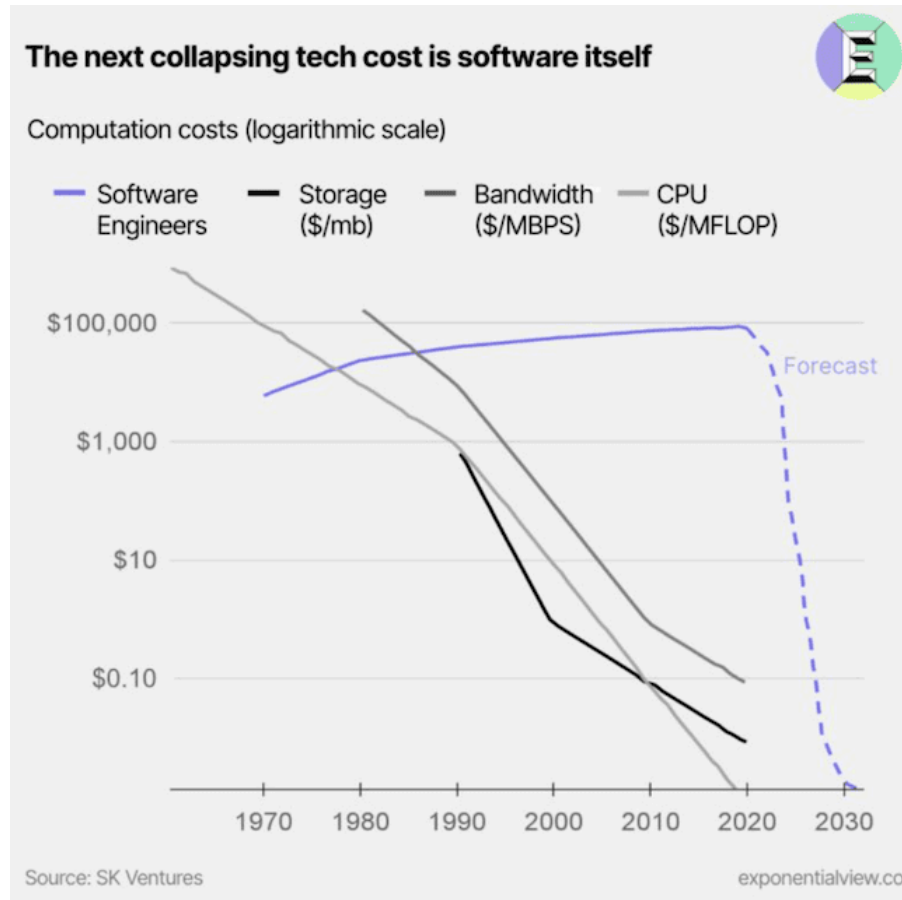


What we have observed recently



- LLMs will handle most coding/testing, with a few exceptions.
- From my experience last year, apps can be built quickly, though ~40% fail without a proper strategy (dumb way of Vibe coding).
- With proper guidance (AI assistive coding), the success rate is 100% with high-quality results.

It looks to be right, but not exactly.



We have some hints from the Stanford report

- Early-career (ages 22-25) in AI-exposed fields (e.g., software development, customer service) have seen a 13% relative decline in employment since late 2022.
- More experienced workers in these same fields have stable or **growing** employment.

We have new findings in SWE

- We don't need junior-level software engineers who can just code.
- AI is the best for prototypes and MVPs (Most Viable Products).
- After we finish the MVP, skilled human efforts are needed to make successful products.

- Due to hallucinations, AI lies frequently; the software engineers who cannot detect the lies will be in deep trouble.
- **Vibe coding** is fun, but misses pro-level details
- Engineers can **start fast**, AI teaches the tech

New Opportunities

How lucky for software engineers that coding/programming is a small part of Software Engineering!

- This is a known fact already in the 1960s, by Fred Brooks, who is considered the father of software engineering, and many others.

Software engineering is about managing complexity

- I knew it from my experience as a professional software engineer, and I found it is a well-known fact (common sense) among researchers during my PhD research.
- Since I worked at NKU in 2017, this has been my main topic of software engineering education.

Software engineering as problem solving by managing complexity

- I re-designed CSC 440 to teach this idea through practices and projects.
- When the ASE program started, I designed ASE courses to teach students problem solving through 4Ps with my colleagues (two research papers published).

NKU ASE 4P4M Approach

4P

- Principles/Patterns
- Practices/Projects

4M

- Maginc/Machine
- Master/Make

My Approaches/Efforts

- Reversing the 4P orders
- Let students know the fun
- Let students ask "whys"
- Make students proud and be known
- Let the industry know what our students can do

Approach 1: Reversing the 4P orders

- I thought students need to master coding to be successful in projects.
- So, I started with "projects/patterns" before "practices/projects".

- It doesn't have to be that way anymore.
- They will learn from AI anyway if they need to learn anything.
- I can help them find solutions effectively, instead of giving them the solutions.

New Findings

- I also found that students can submit the homework answers using AI instantly.
- I know that some students will use AI for cheating in the midterms.

I couldn't prevent them from doing so even when AI was not available, so I may need to change their focus.

So, the focus is now on **projects**: students do the project and learn principles by doing projects. They do homework/midterms to help them learn the knowledge for projects.

efforts

Rewriting my ASE courses

- It was coding intensive (50%), project only after the coding part is done (about week 10)
- I changed it to a practice/project-focused start, starting the project in week 4.

Rewriting all the
content/homework/midterms

- They begin with the knowledge to start the project.
- They learn by doing.
- If they need to know anything, they can ask me or ask AI.

Approach 2: Let students know the fun of making

- We all know making is fun!
- It is even better to make money out of the fun of making (as a professional software engineer).

efforts: Students do two projects:
one team project & one individual
project

- In the team project, they make an interesting product in a team by following rules and using tools.
- Team leaders learn how to manage people.
- Team members learn how to work with others

- In the individual project, they make the software that makes them feel fun.
- It can be anything, including research/product-oriented projects, as long as it fits the goal of the course.

For example, students can make a cross-platform application "Learning Design Patterns" in ASE 420/456

Students have three options

1. Option 1: No usage of AI
 2. Option 2: AI assistive programming
 3. Option 3: Vibe/Low Coding
- Each student makes clear what option they will choose for their project.
 - They don't need to hide or cheat; they do their best to get the best results using the tool they choose

Approach 3: Let students ask "whys" and find the answers

- The industry doesn't want any students who can make programs.
- The industry wants students who can get solutions to the given problems.
- We call this "research" and it starts with asking "why".

Efforts: I encourage students to do research using the course materials and individual projects.

- They can start research that they are curious about or want to know
- AI will guide them to find answers once they ask correct questions:
"why"

- It can be any new area, or students in other fields can use ASE courses to learn how to solve their problems.

Approach 4: Make students proud and be known

- When they finish their projects, they should feel proud.
- In this way, they will be eager to solve the next-level problems in other ASE courses.
- They also add more lines in their resume & portfolio.

- Their LinkedIn does not have the course that they took, but the projects (and GitHub links) that they finished.
- Their results are known in a variety of formats and they feel proud.

- At some point in time (or iterations), they may be able to get the senior-level problem-solving skills in this way.
- From my experience, some students will get this level of skills with the help of AI quickly.

It may be a daydream, but what other options or approaches do I have?

efforts:

- Celebration
- Any form of student research support, including App development or IoT system build-up.
- Any form of demonstration
- YouTube video clips

Approach 5: Let the industry know what our students can do

- The company that I worked for has an interesting tradition that is very successful.

- They hire interns only from the universities that the successful software engineers are from.
- They work as team member with the alums already.
- The company keeps contacting and offering jobs.

- Maybe we can make similar efforts.
- After we make students good/competent problem solvers, we need to let the companies know that.

efforts:

- Hackathon: We will have the "NKU Hackathon" next week!
 - It is about the finance sector (FIN Tech).
- More contact points
- Let companies offer to work on the problems they need to address
- Using the alum network actively

My Other Efforts

I need your input/feedback.

A new way of software development courses

- Making new courses that teach students how to use AI to solve problems in a specific domain.

CSC 494: Use AI to learn complex topics of IoT is an example

- Delegating coding/programming to AI so we can focus on problem-solving
- Maybe, Applied AI - Problem Solving with AI
- Applied AI - software system building with AI

A member of the AI initiative

- I want to hear about others' experiences.
- Maybe, Vibe coding or AI assistive coding practices/seminars?

More Hackathons?

- IoT Hackathon
- AI Hackathon

Some students may come up with the idea for an individual project/research in various areas.

Thanks

Let's find the optimal solution
together to take advantage of this AI
revolution!

Visit me if you'd like me to show the
demos or discuss topics (I have many
more)!