

---

# Introduction

SWE233: Intelligent User Interfaces

Daye Nam  
Fall 2025

---

# Introductions

# Daye Nam

B.S. Yonsei University 2016

M.S. University of Southern California 2018

Ph.D. Carnegie Mellon University 2024

Research Scientist at Google 2024-2025

Assistant Professor at UC Irvine

# You?

Your (preferred) name

Why are you taking this course?

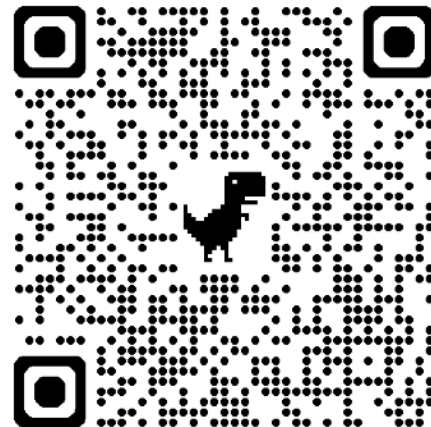
A fun fact

---

# Activity

Intro survey

<https://forms.gle/EJJq5qGcSzhPSZV96>



---

# Intelligent User Interfaces

# Intelligent User Interfaces

## **User interface**

Everything the user encounters when using a product (e.g., a particular machine, device, computer program, or system)

## **Intelligent user interface**

User interface that is intelligent

Intelligent user interfaces can

- complement human capabilities by identifying and augmenting shortcomings in what people can do;

- be integrated into the tools that people and organizations already use to structure and carry out their work

## Questions we will ask

How can interaction be made clearer and more efficient?

How can interfaces offer better support for their users' plans and goals?

How can information be presented more clearly and effectively?

How can the design and implementation of good interfaces be made easier?

...



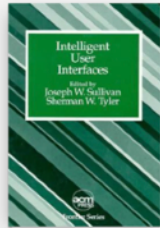
# Questions that have been asked for ~50 years

How can interaction be made clearer and more efficient?

How can interfaces offer better support for their users' plans and goals?

How can informatino be presented more clearly and effectively?

How can the design and implementation of good interfaces be made easier?



CONFERENCE PROCEEDING

Intelligent user interfaces / edited by Joseph W. Sullivan, Sherman W. Tyler.

Sullivan, Joseph W. (Joseph William), 1947-; Tyler, Sherman W.

©1991

Available at [Science Library \(QA76.9.H85 I583 1991\)](#) >

- "... It addresses the same questions that have driven research in the general area of user interfaces for the past ten years."

# ELIZA



Created by Joseph  
Weizenbaum in 1964

One of the first natural  
language interfaces

# Knowledge Navigator



Apple concept video from 1987

Created by Hugh Dubberly et al., for a keynote speech of CEO John Sculley

# Visily



# Google Glass (old version)



# Answers to the questions so far

## **How can interaction be made clearer and more efficient?**

ELIZA, by supporting a natural language-based interaction and minimizing the necessary domain knowledge (e.g., programming) and cognitive load

# Answers to the questions so far

## **How can interaction be made clearer and more efficient?**

ELIZA, by supporting a natural language-based interaction and minimizing the necessary domain knowledge (e.g., programming) and cognitive load

## **How can interfaces offer better support for their users' plans and goals?**

Knowledge navigator, by acting as a proactive, intelligent agent that understands high-level, conversational intent and automates the complex workflows required to achieve those goals.

# Answers to the questions so far

## **How can interaction be made clearer and more efficient?**

ELIZA, by supporting a natural language-based interaction and minimizing the necessary domain knowledge (e.g., programming) and cognitive load

## **How can interfaces offer better support for their users' plans and goals?**

Knowledge navigator, by acting as a proactive, intelligent agent that understands high-level, conversational intent and automates the complex workflows required to achieve those goals.

## **How can information be presented more clearly and effectively?**

Google glass, by integrating information directly into the user's field of view and reducing the cognitive load required to switch between a physical task and a separate screen



# Answers to the questions so far

## **How can interaction be made clearer and more efficient?**

ELIZA, by supporting a natural language-based interaction and mimicking the necessary domain knowledge (e.g., programming) and cognitive load

## **How can interfaces offer better support for their users' plans and goals?**

Knowledge navigator, by acting as a proactive, intelligent agent that understands high-level, conversational intent and automates the complex workflows required to achieve those goals.

## **How can information be presented more clearly and effectively?**

Google glass, by integrating information directly into the user's field of view and reducing the cognitive load required to switch between a physical task and a separate screen

## **How can the design and implementation of good interfaces be made easier?**

Visibly, by automatically converting a static image into a fully editable mockup and reducing the user loads

# Course Themes

- Foundations of Human-Computer Interaction and UI Design Principles
- User Modeling and Personalization Techniques
- Adaptive and Context-Aware Interfaces
- Conversational AI and Natural Language Interfaces
- Agentic Interfaces and Multi-Step Task Delegation
- Multi-Modal Interfaces (Voice, Gesture, AR/VR)
- Programming by Demonstration and Example-Based Systems
- Research Methods for HCI (Surveys, Interviews, Qualitative Analysis)
- Evaluation Methods for Intelligent UI (A/B Testing, Longitudinal Studies, Wizard of Oz)
- IDEs and AI-Powered Programming Tools
- Debugging and Search Interfaces
- Data Science Workflows and Computational Notebooks
- Explainable AI (XAI) in User Interfaces
- Trust, Transparency, and User Control in Intelligent Systems

---

# Course logistics

# Evaluation

## **80% Research Project**

**5%** Problem Identification

**25%** Project Proposal

Report: 15%

Presentation: 10%

**15%** Prototype Design

**35%** Final Report

Report: 15%

Presentation: 10%

Holistic evaluation: 10%

**10%** Paper Presentation

**10%** In-Class Participation

**[Optional] 10%** Prototype Implementation

# Research Project

A quarter-long project, where you design and prototype an intelligent user interface.

Expected outcome:  
A CHI/IJIST/ICSE/FSE/ACL/Neurips...  
paper except for the results  
Abstract  
Introduction  
Literature review  
Methods  
Evaluation

...

## SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering

John Yang<sup>1</sup> Cu  
Shanyu  
Princeton

Language model (LLM)  
tools in digital design  
applications, such as  
software engineering,  
users with their own  
interfaces to the software  
performance of language  
SWE-agent: a system  
solve software engine  
(ACE) significantly out  
perform existing tools,  
on SWE-bench and 10  
with a pass@1 rate of  
state-of-the-art achieve  
how the design of the

### "What It Wants Me To Say": Bridging the Abstraction Gap Between End-User Programmers and Code-Generating Large Language Models

Michael Xieyang Lin<sup>1</sup>  
Microsoft Research,  
Carnegie

Adwait Sarkar<sup>1</sup>  
Microsoft Research,

Carina Negreanu  
Microsoft Research

Bert  
Mier

### Using an LLM to Help With Code Understanding

Dave Nann  
Google, Inc.,  
U.S.A.  
dann@cs.cmu.edu

Andrew Macevan  
Google, Inc.,  
U.S.A.  
amacevan@google.com

Vincent Hellendoorn  
Carnegie Mellon University  
U.S.A.  
vhellend@andrew.cmu.edu

Regdan Vasilescu  
Carnegie Mellon University  
U.S.A.  
vasilescu@cmu.edu

Brad Myers  
Carnegie Mellon University  
U.S.A.  
bmyers@cs.cmu.edu

#### ABSTRACT

Code-generating LLMs  
code. However, end  
users often struggle to  
non-expert end-user  
context of data and  
the user's natural  
generator, executor  
grounded abstraction  
by translating the  
a realistic interface. In  
we compare growth  
tentative based on  
We find that the  
standing of the way  
and the kind of help

#### CCS CONCEPTS

Human-centered  
Interactive systems

#### KEYWORDS

Natural Language  
User, Large Language

#### INTRODUCTION

Building and maintaining  
software systems requires  
a deep understanding  
of a codebase. Consequently,  
developers spend a sig-  
nificant amount of time  
searching and finding for  
the information they  
need and organizing and  
digesting the information

#### ABSTRACT

Understanding code is  
challenging, especially  
when working in new  
and complex development  
environments. Code com-  
ments and documentation  
can help, but are typically  
sparse or hard to navigate.  
Large language models  
(LLMs) are revolutionizing  
the process of writing  
code. Can they do the same  
for helping understand it?  
In this study, we provide a  
first investigation of an LLM-  
based conversational UI  
built directly in the IDE that  
is geared towards code  
understanding. Our IDE  
plugin queries OpenAI's GPT-3.5-  
turbo model with four high-  
level requests without the  
user having to write explicit  
prompts: to explain a high-  
lighted section of code,  
provide details of API calls  
used in the code, explain key  
domain-specific terms, and  
provide usage examples for  
an API. The plugin also  
allows for open-ended  
prompts, which are automati-  
cally translated to the LLM  
with the program being  
edited. We evaluate this  
system in a user study with  
32 participants, which con-  
firms that using our plugin  
can aid in tasks completion  
more than web search. We  
additionally provide a  
thorough analysis of the  
ways developers use, and  
perceive the usefulness of,  
our system, among other  
findings that the usage and  
benefits differ between  
strains and professionals.  
We conclude that to-  
day's prompt-less interaction  
with LLMs is a promising  
future direction for tool  
builders.

#### ACM Reference Format

Dave Nann, Andrew Macevan, Vincent Hellendoorn, Regdan Vasilescu, and Brad Myers. 2024. Using an LLM to Help With Code Understanding. In *36th IEEE/ACM 44th International Conference on Software Engineering (ICSE '24)*, April 14–20, 2024, Lisbon, Portugal. ACM, New York, NY, USA, 23 pages. <https://doi.org/10.1145/3659713.3659717>

#### 1 INTRODUCTION

Building and maintaining  
software systems requires  
a deep understanding  
of a codebase. Consequently,  
developers spend a sig-  
nificant amount of time  
searching and finding for  
the information they  
need and organizing and  
digesting the information



This work is licensed under a Creative Commons  
Attribution-NonCommercial 4.0 International  
License.  
ICSE '24, April 14–20, 2024, Lisbon, Portugal.  
© 2024 Copyright held by the owner(s).  
ACM ISBN 978-1-4503-9973-5/24/0000.  
<https://doi.org/10.1145/3659713.3659717>

## Project (Tentative timeline)

**Oct 8: Problem Identification** Identify a problem for intelligent UI solution

**Oct 27: Project Proposal** Literature review and proposed solution

**Oct 20&22: Proposal Presentation** Present your proposal for feedback

**Nov 17: Prototype Design** System architecture and interface design

**Dec 10: Final Report** Complete evaluation methodology

**Dec 1 & 3: Final Presentation** Showcase your complete project

# Paper Presentation & Discussion

We will read and discuss systems with intelligent user interfaces, mainly within the software engineering domain

## **Presentation**

Each student will present at least one paper over the quarter.

As presenters, you will do a presentation of the paper and connect the paper to many aspects we cover in the first part of the quarter.

## **Paper selection**

You will bid for a paper you want to present in the next class.

Bid for papers that are close to your potential project topic.

# Course infrastructures and logistics

## Infrastructure/source of truth

Course website: schedule, slides, syllabus, office hours

Canvas: homework, grades, other material

Slack: communication and collaboration (will send you today)

## Logistics

Lecture in-person only

You are required to attend presentation sessions (proposal/final)

Office hours: Monday 4:00 pm - 5:00 pm, but flexible