EECS 22L: Software Engineering Project in C Language

Lecture 7

Rainer Dömer

doemer@uci.edu

The Henry Samueli School of Engineering
Electrical Engineering and Computer Science
University of California, Irvine

Lecture 7: Overview

- Course Administration
  - Midterm course evaluation
  - New project setup
- Project 2
  - Introduction
  - Focus: GUI, CVS, testing
  - Application specification

Course Administration

- Midterm Course Evaluation: Results
  - Participation
    - 36 out of 71 students (50.7%)
    - ➢ Somewhat representative
  - Student Feedback
    - Mostly very positive and encouraging
      - Overall letter grade: 19 “A”, 13 “A-”, 3 “B+”
    - Specific comments
      - ➢ **`MidtermEvaluation_Results.pdf`**

Course Administration

- New Teams
  - New project 2 teams have been formed
    - Most preferences have been met
      - Some teams stay together unchanged
    - Very few exceptions (cyclic conflicts)
  - 10 new teams, 7 to 8 members each
    - See individual team emails
- New Team Accounts
  - All team accounts have been cleared (all files deleted)
  - New passwords will be distributed in discussion session
    - Use lab sessions this week to set up fresh team account

Project 2

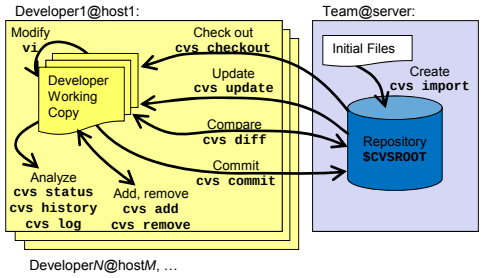
- New Focus
  - Graphical User Interface (GUI)
    - mandatory
  - Version control with CVS
    - mandatory
  - Testing
    - Unit test
    - System test

Project 2 Focus: GUI

- Graphical User Interface (GUI)
  - Differences to command-line interface
    - Input through events
    - Output through 2D pixel interface
    - Event-based main control loop
  - Available libraries
    - SDL: Simple DirectMedia Layer
      - **`http://www.libsdl.org/`**
    - GTK: GNOME or GIMP Toolkit
      - **`http://www.gtk.org/`**
  - Refer to Lecture 4 and online documentation

Project 2 Focus: CVS

- Version Control using CVS
  - Source code management
    - For successful team work, it is critical to have the same data at the same state
    - Once set up, it's all a matter of '**`cvs update`**' and '**`make clean all`**'



  - ➢ Refer to Lecture 3 and online documentation

Project 2 Focus: Testing

- Perform Automated Unit and System Test
  - Test each module/component individually
    - With specific test data
  - Run each module/component in debug mode
    - **`make test`**
  - ➢ Top-level **`Makefile`** should provide targets to run tests for each unit and the entire system
  - Unit tests
    - **`% make test_gui`**
    - **`% make test_client`**
    - **`% make test_server`**
  - System test
    - **`% make test`**

Project 2 Focus: Testing

- Perform Automated Unit and System Test
  - Test each module individually (with specific test data)
  - Use **`make test`** to run each module in debug mode
    - Example: Student records (see EECS 22, Lectures 14 ff.)

```
/* Student.c: maintaining student records */
...
#ifdef MAIN  /* test the student record functions */
int main(void)
{   STUDENT *s1 = NULL;
    s1 = NewStudent(1001, "Jane Doe", 'A');
    PrintStudent(s1);
    [...]
    return 0;
} /* end of main */
#endif /* MAIN */
/* EOF */
```

```
% vi Student.c
% make Student
gcc –Wall –ansi –g –c Student.c –o Student.o
gcc -DMAIN -Wall -ansi -g Student.c Student.o -o Student
```


Project 2

- Introduction
  - Taxi Cab Management System
  - Story:

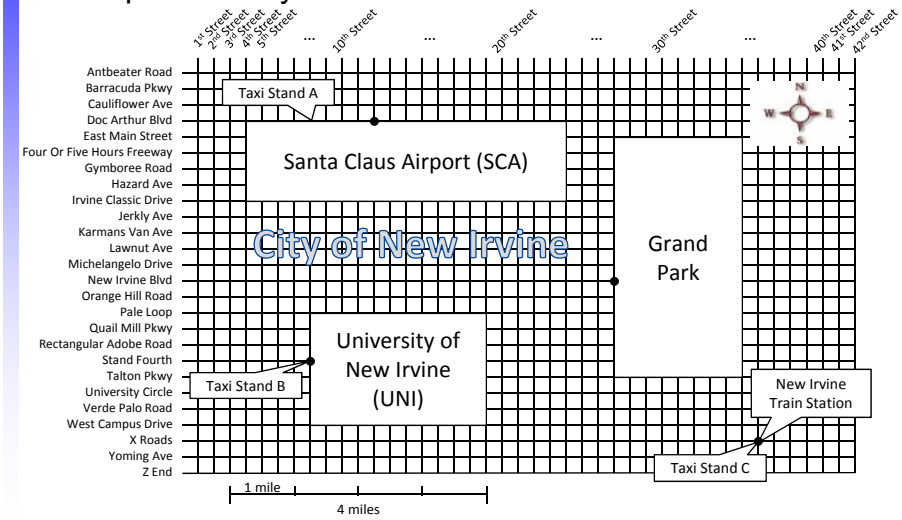
    *The **City of New Irvine** is soliciting proposals for the development of a new fully-automated taxi cab management service that can optimally coordinate a number of taxi cabs serving the individual public transportation needs of the population and visitors of the city.*

    The City of New Irvine expects transportation by taxi to be very flexible and efficient for individual people and small groups. Customers may request rides from and to every street corner within the city limits, to be served at the time of request or at a defined later time as specified by the customer.
  - Task:

    Design a networked computer server where incoming customer requests are automatically processed and taxi drivers can fairly compete for trips and compensation. At the same time, customers' expectations of punctuality and quick trips are to be met to the best extend possible.

Project 2

- Map of the City of New Irvine

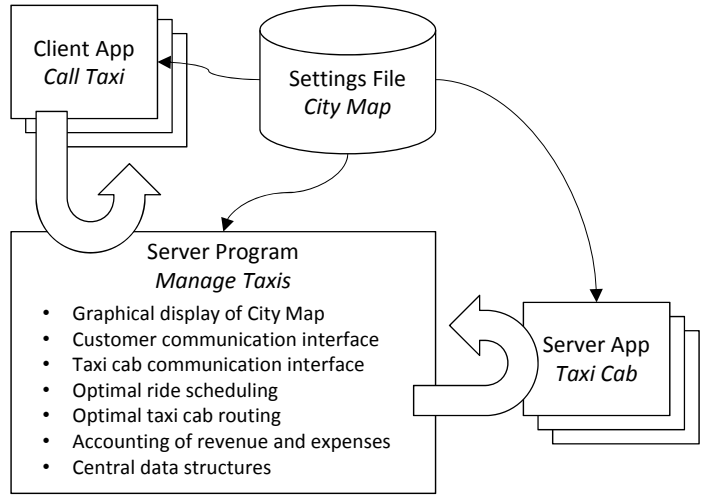


Project 2

- Goal and Requirements
  - Customer rides from/to every corner or landmark
  - Customer rides now or at specified later time
  - Up to about 20 minutes ride, obeying 45 MPH speed limit
  - Taxi stands
    - Up to 12 cab cars
    - Capacity limousine 3 passengers, van 6 passengers
  - Fair and easy transportation of individuals and small groups
  - Punctuality and quick trips
  - Taxi costs regulated
    - Fixed base fee $3.75 (for pickup and return)
    - Manhattan-distance dependent charge of $2.00 per mile
    - Taxi drivers earn $0.20 per block driven (incl. pickup and return)
    - Management system shall generate revenue!

Project 2

- Overall System Specification



Project 2

- System Management Settings
  - Settings file (extendable)

```
STREETS EAST-WEST:
Antbeater Road, Barracuda Pkwy, Cauliflower Ave, Doc Arthur Blvd, East Main Street,
Four Or Five Hours Freeway, Gymboree Road, Hazard Ave, Irvine Classic Drive, Jerkly
Ave, Karmans Van Ave, Lawnut Ave, Michelangelo Drive, New Irvine Blvd, Orange Hill
Road, Pale Loop, Quail Mill Pkwy, Rectangular Adobe Road, Stand Fourth, Talton Pkwy,
University Circle, Verde Palo Road, West Campus Drive, X Roads, Yoming Ave, Z End.

STREETS NORTH-SOUTH:
1st Street, 2nd Street, 3rd Street, […street names omitted for brevity…], 42nd Street.

LANDMARKS:
University of New Irvine (UNI) AT Stand Fourth AND 8th Street BOUNDARIES NORTH Pale
Loop EAST 19th Street SOUTH West Campus Drive WEST 8th Street,
 Santa Claus Airport (SCA) AT Doc Arthur Blvd AND 12th Street BOUNDARIES NORTH Doc
Arthur Blvd EAST 24th Street SOUTH Irvine Classic Drive WEST 4th Street,
Grand Park AT New Irvine Blvd AND 27th Street BOUNDARIES NORTH East Main Street SOUTH
Talton Pkwy EAST 35th Street WEST 27th Street,
New Irvine Train Station AT X Roads AND 36th Street,
Taxi Stand A AT Doc Arthur Blvd AND 8th Street,
Taxi Stand B AT Stand Fourth AND 8th Street,
Taxi Stand C AT X Roads AND 36th Street.
```

Project 2

- Protocol *"Call Taxi"*
  - Request in Backus-Naur Form (BNF)

```
<request> ::= REQUEST RIDE FROM <location> TO <location> <special>*
<location> ::= CORNER <street name> AND <street name>
        | <landmark name>
<special> ::= [<time>]
        | [FOR <number> PERSONS]
        | [FIRST CLASS]
<time> ::= AT <hours>:<minutes>
        | NOW
```

  - Response

```
<response> ::= OK PICKUP AT <estimated time> COSTS
$<dollars>.<cents>
        | DECLINED <reason>
        | ERROR <message>
```


Project 2

- Protocol *"Call Taxi"*
  - Request in Backus-Naur Form (BNF)

```
<message> ::= <request>
        | <instruction>
<request> ::= REQUEST POSITION
<instruction> ::= GO <direction> <direction>*
        | PICKUP <number> PERSONS
        | DROPOFF <number> PERSONS
        | GOODBYE
```

  - Response

```
<response> ::= OK POSITION AT <location>
        | OK FOLLOWING DIRECTIONS
        | OK PICKUP | DROPOFF | GOODBYE
        | ERROR POSITION <message>
        | ERROR DIRECTIONS <message>
        | ERROR PICKUP <message>
        | ERROR DROPOFF <message>
        | ERROR <message>
```