

First Model, then Function — on the value of objects



Online course evaluations are starting

Completion counts toward your course grade like a quiz

Just before the final, we get a list of students who completed it. Don't see actual evaluations until grades have been submitted

Object-oriented design

The foundation of the system should be a model of relevant concepts and phenomena from the problem domain

People at a university. Organize into categories

- university member
 - student
 - grad
 - undergrad
 - nondegree
 - faculty
 - prof
 - ...
 - staff
 - ...

is-a relation

Make B a subclass of C is every instance of B is a C

Subclass principle

Structure classes so that behavior common to several classes can be defined in a superclass of those classes

First Model, then Function — on the value of objects



Gries's revision of lecture by Michael E. Caspersen
University of Aarhus
Denmark

The foundation of the system should be a model of relevant concepts and phenomena from the problem domain

Overview

Function first, or model first?

- One problem
- Two solutions: a bad and a good
- Very black and white
- Exaggeration promotes understanding

The Coming of Showboat Inc., (2)

ITHACA YACHT CLUB
Boat hiring by the minute

Showboat Inc. and Contractors

Daniel von Schneider: own, manage company

Sailor Sam: At the beach, rent out boats

John H. Acker: freelance programmer

Scarlett Olivia Oakley Patton (Scoop): promising employee of software firm

Requirements

- von Schneider requests a decision support system for strategic planning
- The system must be able to deliver a daily report
 - number of sessions (hires) of the day
 - average duration of each session
- von Schneider asks for bids ... John H. Acker wins

Cornell University, CS1110, Spring 2009

Function First



- John H. Acker is smart, very smart!
- Realizes he can get by with a two-variable state space
 - n : number of finished sessions
 - $totalTime$: sum of duration of finished session
- Maintaining n is easy
 - finish a session? Increment n
- Maintaining $totalTime$ is less trivial...

Cornell University, CS1110, Spring 2009

“Clever” Calculations...



s_i : start time for session i
 e_i : end time for session i
 duration of session i : $(e_i - s_i)$

$totalTime$
 = { definition }
 $(e_1 - s_1) + (e_2 - s_2) + \dots + (e_n - s_n)$
 = { remove parentheses }
 $e_1 - s_1 + e_2 - s_2 + \dots + e_n - s_n$
 = { re-ordering }
 $e_1 + e_2 + \dots + e_n - s_1 - s_2 - \dots - s_n$

Start of session: $totalTime = totalTime - currentTime$
 End of session: $totalTime = totalTime + currentTime$

Cornell University, CS1110, Spring 2009

Use of the System



Sailor Sam



von Schneider

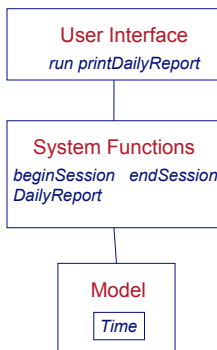
Control panel

Start End Day over

```
*****
*   S H O W B O A T   I n c   *
*****
Daily Report   21 Apr 2008   22:29
Number of Sessions:      5
Average rental time:    00:12:02
*****
```

Cornell University, CS1110, Spring 2009

J. H. Acker’s Software Architecture



Time:
 general class in a class library;
 not domain specific

```

t0
Time
+ Time ()
+ Time (int t)
+ Time add (Time t)
+ Time subtract (Time t)
+ Time divide (int n)
+ ...
    
```

Cornell University, CS1110, Spring 2009

System Functions



```

private int n= 0;
private Time totalTime= new Time(0);
public void beginSession () {
    Time now= new Time();
    totalTime= totalTime.subtract(now);
}
public void endSession () {
    Time now= new Time();
    totalTime= totalTime.add(now);
    n= n + 1;
}
public void doDailyReport () {
    Time avgTime;
    if (n != 0) avgTime= totalTime.divide(n);
    else avgTime= new Time(0);
    printDailyReport(n, avgTime);
}
    
```

Cornell University, CS1110, Spring 2009

User Interface



```
public static void run() {
    char command= '.';
    int sessionNr= 0;

    command= getChar();
    while (command != '.') {
        if (command == 's')
            f.beginSession();
        if (command == 'e')
            f.endSession();
        command= getChar();
    }
    printDailyReport(f.doDailyReport());
}
```

Cornell University, CS1110, Spring 2009

Two Months Later



- von Schneider is pleased with system but quickly gets new ideas...
- von Schneider asks John H. Acker to make a few harmless extensions to the daily report to make it more ... interesting:
 - duration of the longest session of the day :-)
 - an extra report at noon :-)
 - distribution of load during the day :-)
 - ... :-) :-) :-)
 - the maximum number of concurrent sessions :-)

John H. Acker Rules No More

Cornell University, CS1110, Spring 2009

Scarlett Olivia Oakley Patton's Object-Oriented Perspective



- John H. Acker's system: developed solely by focusing on functional requirements of the system
- No software representation of concepts and phenomena in the problem domain
- Foundation of system should be a model of relevant concepts and phenomena from the problem domain
- **Relevance**: determined by the requirements specification
- But what are the relevant concepts...?

Cornell University, CS1110, Spring 2009

The Key Concept: Session



- Relevant concepts: found in requirements spec (functional requirements)
 - no. **sessions** (hires) per day
 - average **session** duration
- Both requirements expressed in terms: **session**
- Central property of a session **duration**
- Must be able to **start** and **stop** a session
- Must be able to **identify** sessions (unique id)

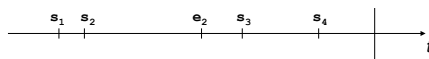
```
class Session {
    /** Constr: session with id n,
     * start/stop times of 0 */
    public Session (int n)
    /** = session id */
    public int id()
    /** Start session now */
    public void start()
    /** End session now */
    public void stop()
    /** = duration of session.
     * Pre: session is stopped */
    public Time duration()
}
```

Cornell University, CS1110, Spring 2009

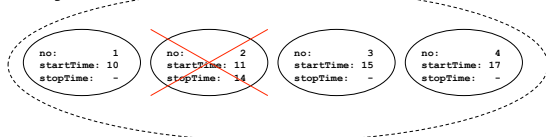
Session Objects



A session object for each unfinished session:



Bag current:



Cornell University, CS1110, Spring 2009

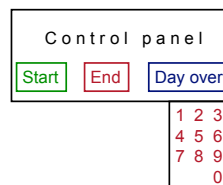
Use of the System



Sailor Sam



von Schneider

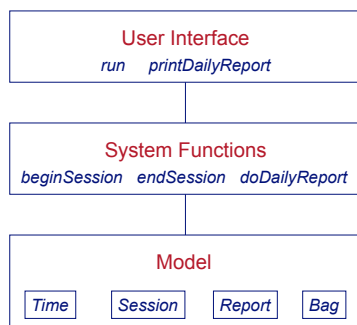


```
*****
*   S H O W B O A T   I n c   *
*****
Daily Report   21 Apr 2008   22:29

Number of Sessions:      5
Average rental time:    00:12:02
*****
```

Cornell University, CS1110, Spring 2009

Scarlett's Software Architecture



Cornell University, CS1110, Spring 2009

User Interface

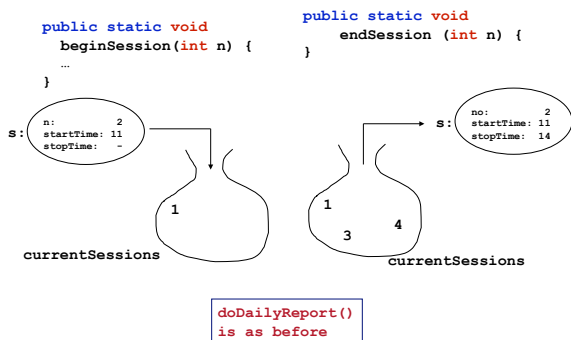
```
public static void run() {
    char command= '.';
    int sessionNo= 0;

    command= getChar();
    while (command != '.') {
        switch (command) {
            case 's': case 'S':
                sessionNo= getNextSession(sessionNo); break;
            case 'e': case 'E':
                f.setInt(); f.endSession(sessionNo); break;
        }
        command= getChar();
    }
    printDailyReport(f.doDailyReport());
}
```

As before (almost)

Cornell University, CS1110, Spring 2009

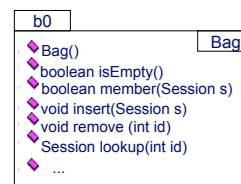
System Functions



Cornell University, CS1110, Spring 2009

Bag

Bag is a general class in a class library; like Time, it is not a domain specific class



An object, from pt of view of the user

Cornell University, CS1110, Spring 2009

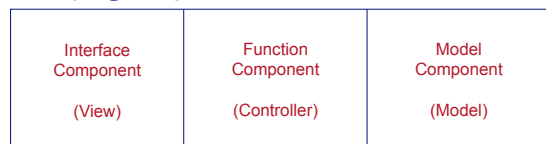
Two Months Later

- von Schneider is pleased about the system but quickly starts getting new ideas...
- Calls Scarlett OOP and asks her to make a few harmless extensions to the daily report to make it more ... interesting:
 - duration of the longest session of the day :-)
 - an extra report at noon :-)
 - distribution of load during the day :-)
 - ... :-) :-) :-)
 - the maximum number of concurrent sessions :-)

Cornell University, CS1110, Spring 2009

Scarlett Olivia Oakley Patton Still Rules

A (logical) Three Tier Architecture



← more dynamic → more static

Cornell University, CS1110, Spring 2009