Overview of Detailed Design

Hui Chen [a]

[a]CUNY Brooklyn College, Brooklyn, NY, USA

April 8, 2025

Outline

Outline

Software Design

- Design starts mostly from/with requirements – evolving mostly from functionalities and other non-functional characteristics
  - In the waterfall model design generally occurs after requirements
  - In agile, design is performed during each iteration
- To answer: How is the software solution going to be structured?
  - What are the main components – (functional composition) often directly from requirements' functionalities (e.g., use cases, user stories, scenarios, storyboards)
  - How are these components related? – Possibly re-organize the components (composition/decomposition)
- Two main levels of design:
  - Architectural (high level) design
  - Detailed design
  - Different design concerns at different abstraction levels (e.g. classes vs. modules vs. entire system)
- How should we depict design – what notation/language?

Outline

Detailed Design

Further refine architecture and match with requirements.

- How detailed?
- How formal?
- Maybe of different levels of detail for different views.

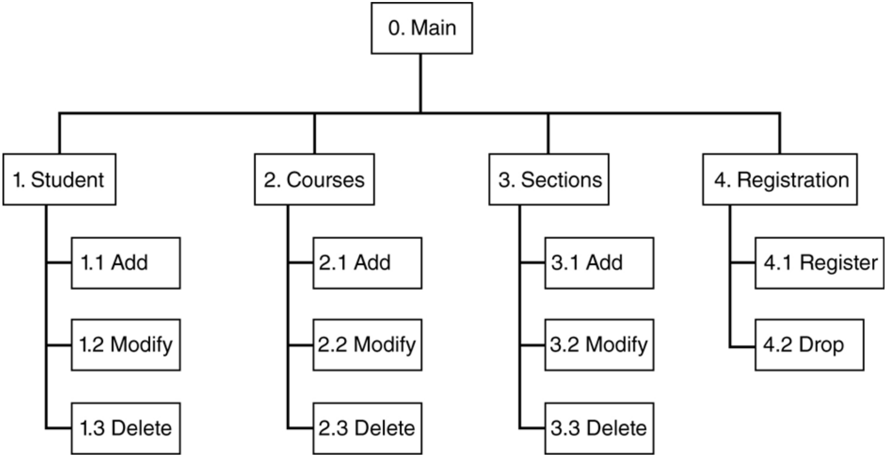
Outline

Functional Decomposition

- Dates back to “structured programming”
- Start with: main (task/requirements) $\rightarrow$ module.
- Refine into sub-modules.
- There are alternative decompositions.

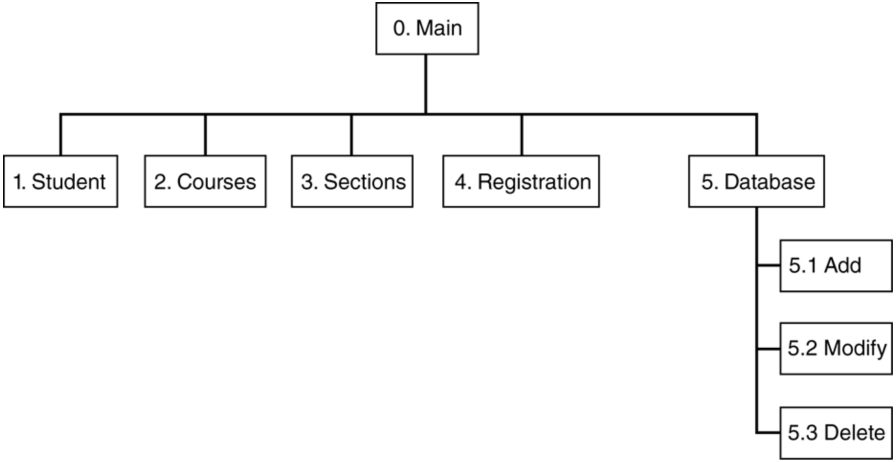
Functional Decomposition: Example 1

Consider a course-student management application (like part of CUNY First).



Functional Decomposition: Example 2

Consider a course-student management application (like part of CUNY First).



Database Design

Many applicationns require a database.

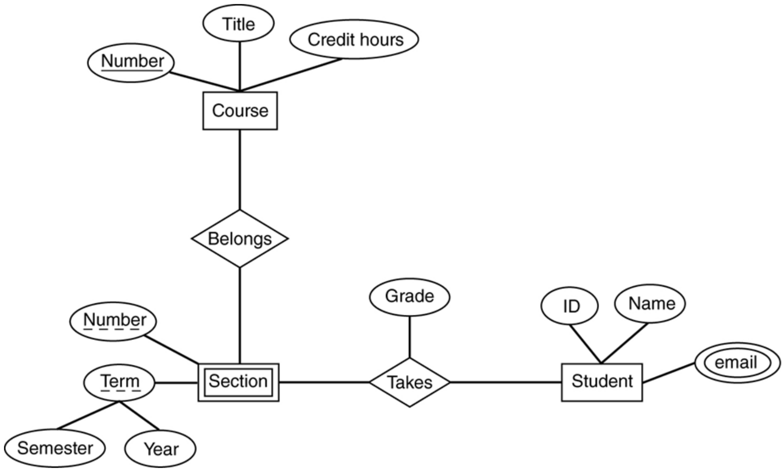
1. Conceptual modeling (done during analysis/requirement phase) produces ER diagram.
2. Logical design (to an actual type of database, e.g., to relational DB).
3. Physical design (decide data types, etc.).
4. Deployment/maintenance
   - Low-level physical (which hard drive, etc.)
   - Adjustment of indexes

Entity-Relationship Model

Typically represented as Entity-Relationship diagrams

- Entities (rectangles)
- Relationships (diamonds)
- Weak: double lines
- Attributes (ovals)
- Multi-valued: double lines
- Identifying (keys): underlined

Entity-Relationship Model: Example



Relational Database Design

Most applications use relational databases.

- Relations (or tables)
  - Two-dimensional sets or bags
  - Rows (tuples), columns (attributes)
  - A row may be an entity; columns may be relationships or attributes.
- Constraints
  - Primary key (unique identifier) – for search
  - Foreign keys (connects tables)
  - . . .

Physical Database Design

- Determine data types for each attribute.
  - Check which ones your DBMS support.
  - Encoding.
- Decide on indexes – data structures stored as part of the database to speed up searches, but also consider:
  - if searches are faster, updates might be slower;
  - indexes consume space; and
  - can always adjust during deployment.
- Denormalization done sometimes (avoid unless you must).

Object-Oriented Design

To be continued ....

User-Interface Design

- Can be part of requirement analysis and elicitation or part of design specification
- In the Agile model (for the class project), as part of the requirements

Most apparent to the user and the concerns are

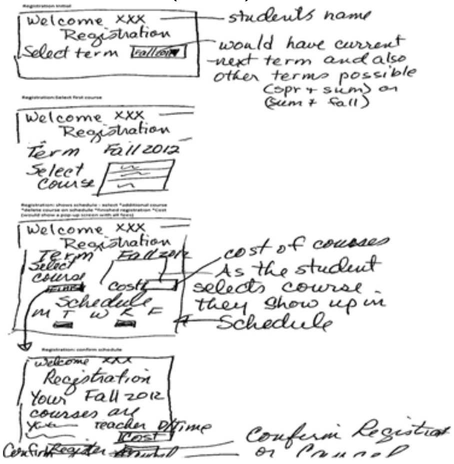
- Two main issues:
  - Flow of interactions
  - Look and feel
- Types of interfaces
  - Command-line
  - Text menus
  - Graphical (GUI)

Flow of Interactions

Recall: sketches and user stories, a low-fidelity (Lo-FI) prototype

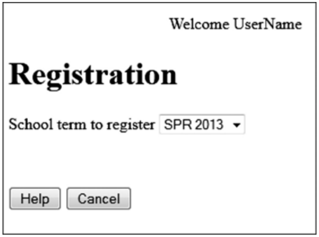
Prototype screens

1. On Registration, select term.
2. Registration shows term, and on it select first course.
3. Registration shows term, course(s) with schedule and cost. On it select among Additional course, Delete course, Finish registration.
4. Registration shows final schedule. On it, select Confirm or Cancel.



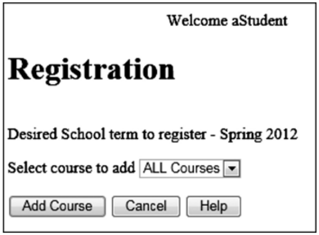
Low-Fidelity vs. High-Fidelity Prototypes

Example of high-fidelity prototypes


Welcome UserName

Registration

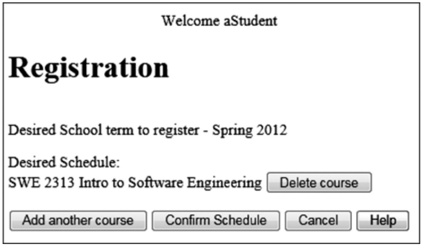
School term to register SPR 2013


Welcome aStudent

Registration

Desired School term to register - Spring 2012

Select course to add ALL Courses


Welcome aStudent

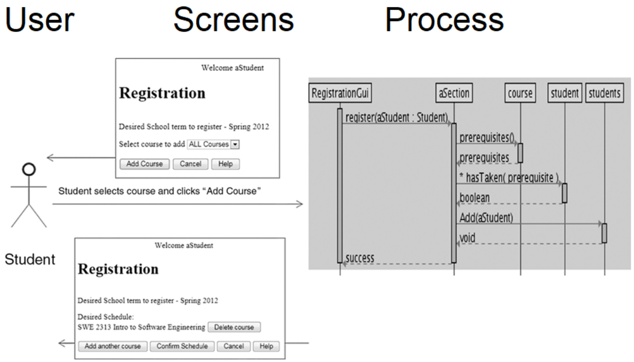
Registration

Desired School term to register - Spring 2012

Desired Schedule:
SWE 2313 Intro to Software Engineering

Flow of Interactions: The Details

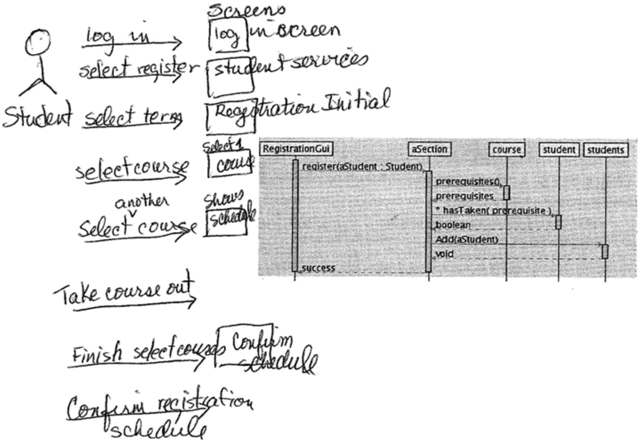
Consider the flow of all possible interactions – which include the user input, the screens, and the process



Add user interactions to the sequence diagram

Flow of Interactions: The Details

Add user interactions to the sequence diagram

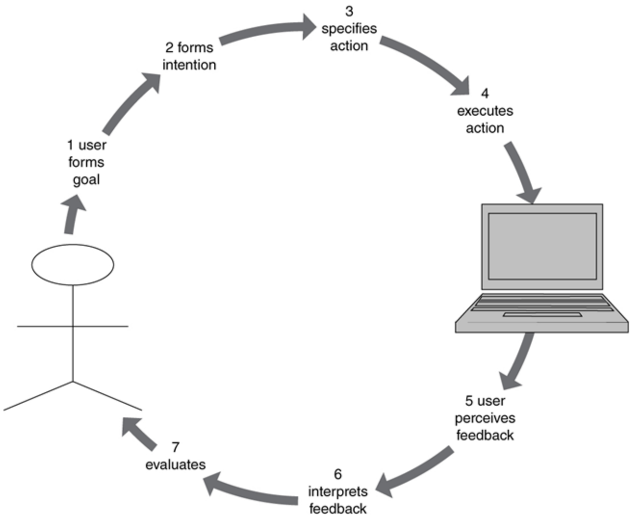


Models for UI Design

Many models are developed ... (an advanced topic for UI design)

- Norman's 7 Stage Model
- The GOMS Model

Norman's 7 Stage Model



Norman's 7 Stage Model

Users will

1. form a goal;
2. form an intention;
3. specify an action;
4. execute the action.
    - After the users execute the action, the feedback from the system is critical for their understanding of the system.
5. Users will perceive the system state (feedback);
6. interpret the feedback; and
7. evaluate.

If, at the last stage, the users evaluate the mode as "intuitive," they will continue with the next cycle toward their goal.

The GOMS Model

Consider different kinds of users.

Four factors (for the kind of user)

- Goals of the user
- Operations provided by the system
- Methods or the sequence of operations
- Selection rules for the methods

Other UI Issues

- Kinds of users
- Heuristics
- UI guidelines
- Multicultural issues
- Metaphors
- Multiplatform software
- Accessibility
- Multimedia interfaces

Summary and Questions

Aspects of detailed design

- Functional decomposition
- Database design
- Object-Oriented design and UML (discuss separately)
- UI design (requirements and design specification)

Outline

"Engineering Software as a Service" by Armando Fox and David Patterson (2nd Edition)
"Essentials of Software Engineering" by Frank Tsui, Orlando Karam, and Barbara Bernal(4th Edition)