Conceptualization

Software Engineering Lab.

Objectives

- Conceptualization
  - *To be able to:*
    - Define conceptualization
    - Describe the conceptualization process

Discussion Question

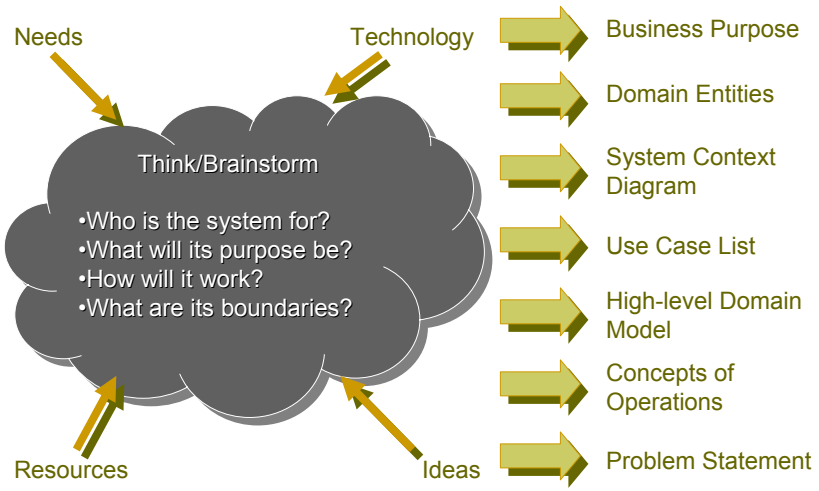
? What is the most important part of building a system?

Building the Right System!

- Attributes Type: A specification of the external behavior and/or the implementation of the attribute
- Consider all possible approaches
  - Each one has a different value to your customer
- Find the approach that best meets the business goals and has the best Return On Investment (ROI)

System development is the gradual eliminating of possible solutions to realize the best system

Conceptualization Process

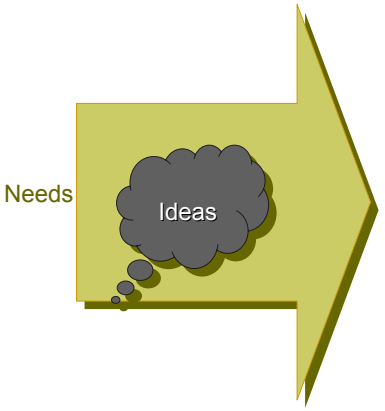


Think/Brainstorm

- Prior to analysis, the problem must be defined
- Knowledgeable people should conceptualize the problem
  - Domain expertise
  - Real-world knowledge
- Keep an open mind
  - Initially, consider all new ideas
  - Later, consider practical aspects such as feasibility, cost, and demand

Brainstorm Your Needs

- Example: Payroll System



Pay employees promptly

Unify the hotel and restaurant systems

Keep track of taxes

Determine current and expected payroll expenses in near real-time

Track level of expertise of employees (training, time on job)

Track employee-incurred expenses

Develop System/Business Purpose

- Concise statement of why the system is needed
  - Shorter cycle time
  - Lower cost
  - Improve quality
  - Increase business productivity
  - Increase value to customer
  - Infrastructure and technology change
  - Expand market

Write Business Purpose Statement

- Example: Payroll System

  Create a payroll system for our company. Make certain that the usual deductions are taken into consideration. The new payroll must accommodate both salaried and hourly employees. The payroll system must issue checks weekly. The system will provide an audit trail. Income and tax reports (such as W-2s* in the U.S.) must be prepared according to the latest legal requirements. Timely reports concerning voluntary deductions must be prepared for various agencies. System must be highly available to employees and their managers. System must be inexpensive to run.

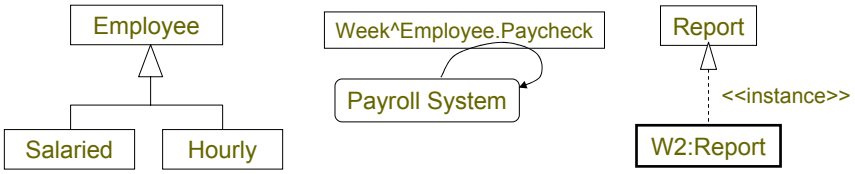
  *W-2 = U.S. annual income statement.

Identify Domain Entities

- Capture key information from the real world
- Identify domain entities and their relationships:
  - Involve subject matter experts in discussions about the system as early as possible
  - Provide a context for conversations with the user
  - Provide consistency within the team
  - Develop a uniform language and terminology for further work
- Use UML modeling notation to represent and explore fundamentals (only consider the domain entities and their relationships)
- Keep conceptualization visual, precise, and consistent

Model Pieces

- Use UML to capture facts about the domain
  - Interview domain experts to obtain information
  - Emphasize clarity, communication. Don't worry about integrating model pieces at this stage



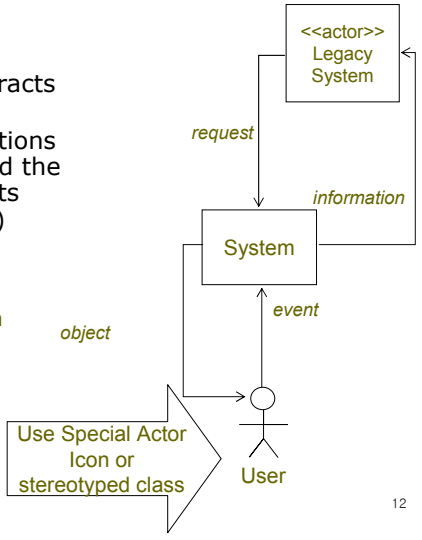
Develop System Context Diagram

- Think of system as a black box
- Establish boundaries for system
- Identify *actors* (who or what interacts with the system)
- Identify fundamental communications that occur between the actors and the system (information and/or events that pass in or out of the system)

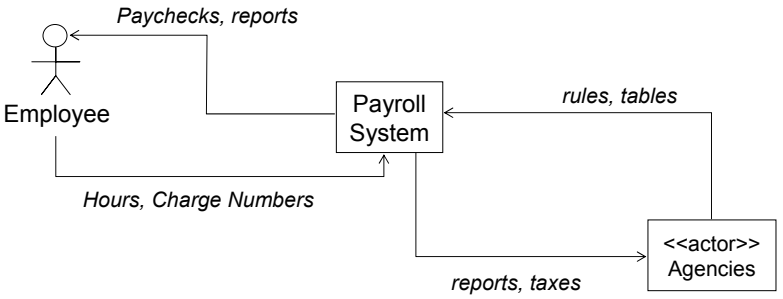
Understand surrounding of the system

and its interaction with the world

- Use standard class diagram form centered around system
- Add associations for each type of communication with the system



Payroll System Context Diagram



Define the Actors

- An actor queries or modifies the system, reports on surroundings, and receives events
- Non-human actors are possible (and likely) in complex systems
  - Machine
  - Another system
  - Clock
- Human actors are the users

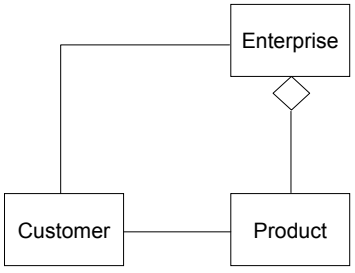
Develop Use Case List

- Ask users what the system needs to do
  - For each actor who uses the system, identify every purpose to which the system will be put
  - Develop a candidate Use Case List

Many answers are possible

Remember : Every separate use of the system requires an initial entry on the list

Develop High-Level Domain Model



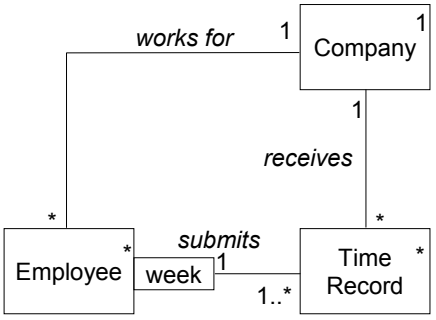
This process drives

(1) Discovery of the essential nature of the enterprise or system

(2) Agreement on basic terminology and representation

- Agree on the names of the key classes, associations, roles
- Determine the multiplicity of the key objects within the system
- Examine each association
  - Determine the multiplicity of each role
  - Look for domain qualifiers and/or association classes
- Most systems start with a three or four class high-level domain model
- May need to develop separate models for each Use Case

Sample High-Level Domain Model



- Is this system for only one company?
- Do employees have to submit weekly time records?
- Can an employee submit more than one time record per week?

Develop Concepts of Operations

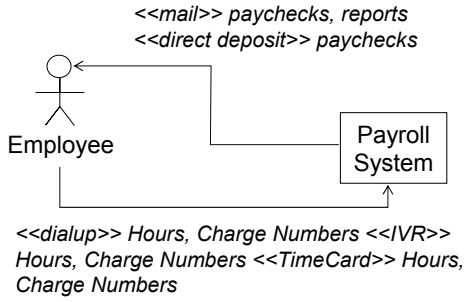
- Determine Concepts of Operations for each Use Case and document with a simple description of:
  - Purpose ~ for what reason will the system be used
  - Approach ~ how it will be done
  - Dynamics ~ when/how often it will be used
  - Goals ~ how will system goals will be targeted

Use Cases & Concepts of Operations Example

- Payroll System
  - Use Cases
    1. Establish relationship with employee (hire, fire, promote…)
    2. Employee reports hours worked and expenses incurred
    3. System pays employee
    4. System writes report
  - Concepts of Operations/Summary for Use Case #2
    - By push-button telephone, timecard
    - Weekly, daily, hourly
    - At personal computer, central terminal, kiosks

Update System Context Diagram

- Update system context diagram to show accepted operations concepts
- Use stereotypes to show different approaches



Write Problem Statement

- The problem statement is a textual description outlining required system capabilities

  - Develop problem statement
    - Incorporate information from the context diagram, Use Case list, and concepts of operations
    - Capture user needs
    - Include performance and functionality requirements
    - Verify that problem statement is correct
    - Analyze capabilities to determine feasibility in terms of technical, cost, and demand
  - Details to be worked out by the analysts and end users

Problem Statement Example

- Create a payroll system for restaurants and hotels. Make certain that the usual deductions are taken into consideration. The payroll must accommodate both salaried and hourly employees. The waiters are salaried employees, but the busboys are hourly employees. The payroll system must print checks weekly. The system will produce a payroll register which will be turned over to auditors monthly. Income and tax reports (such as W-2s* in the U.S.) must be prepared according to legal requirements. Reports concerning voluntary deductions will be prepared for various agencies on a quarterly basis.

- The payroll must treat part-time employees as hourly employees. Full-time employees and salaried employees may take advantage of the various company benefits, part-time employees may not. Restaurant employees will be able to eat meals at their restaurant but will have the cost of the meals deducted from their paycheck. Hotel employees will have room costs deducted if they live in the hotel. There are voluntary deductions and mandatory government deductions that must be taken into account.

  - **W-2 = U.S. annual income statement*

How Much Conceptualization is Required?

- Some systems are well defined by the customer
  - Minimal conceptualization required

- Some systems are only in the customer's head
  - Maximum conceptualization required

Summary

- Conceptualization
  - *Conceptualization* is formulating an idea for a system with a general idea of its requirements and form
  - The conceptualization process consists of:
    - Thinking/brainstorming
    - Develop system/business purpose
    - Develop context diagram
    - Create domain entities
    - Where applicable, identify Use Cases and Concepts of Operations
    - Write a problem statement