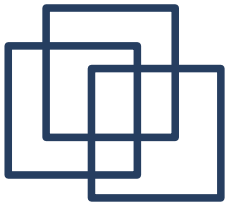
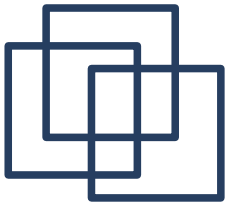


CMSC 128

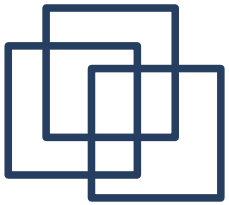
Introduction to Software Engineering
Second Semester AY 2007-2008

jachermocilla@uplb.edu.ph



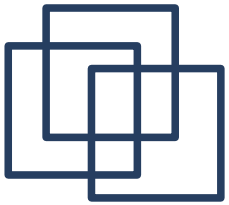
Analysis Modeling

- At the technical level, software engineering begins with a series of modeling tasks
  - Complete specification of requirements
  - Comprehensive design representation
- The analysis model is the first technical representation of a system
  - Structured analysis
  - Object-oriented analysis



Structured Analysis

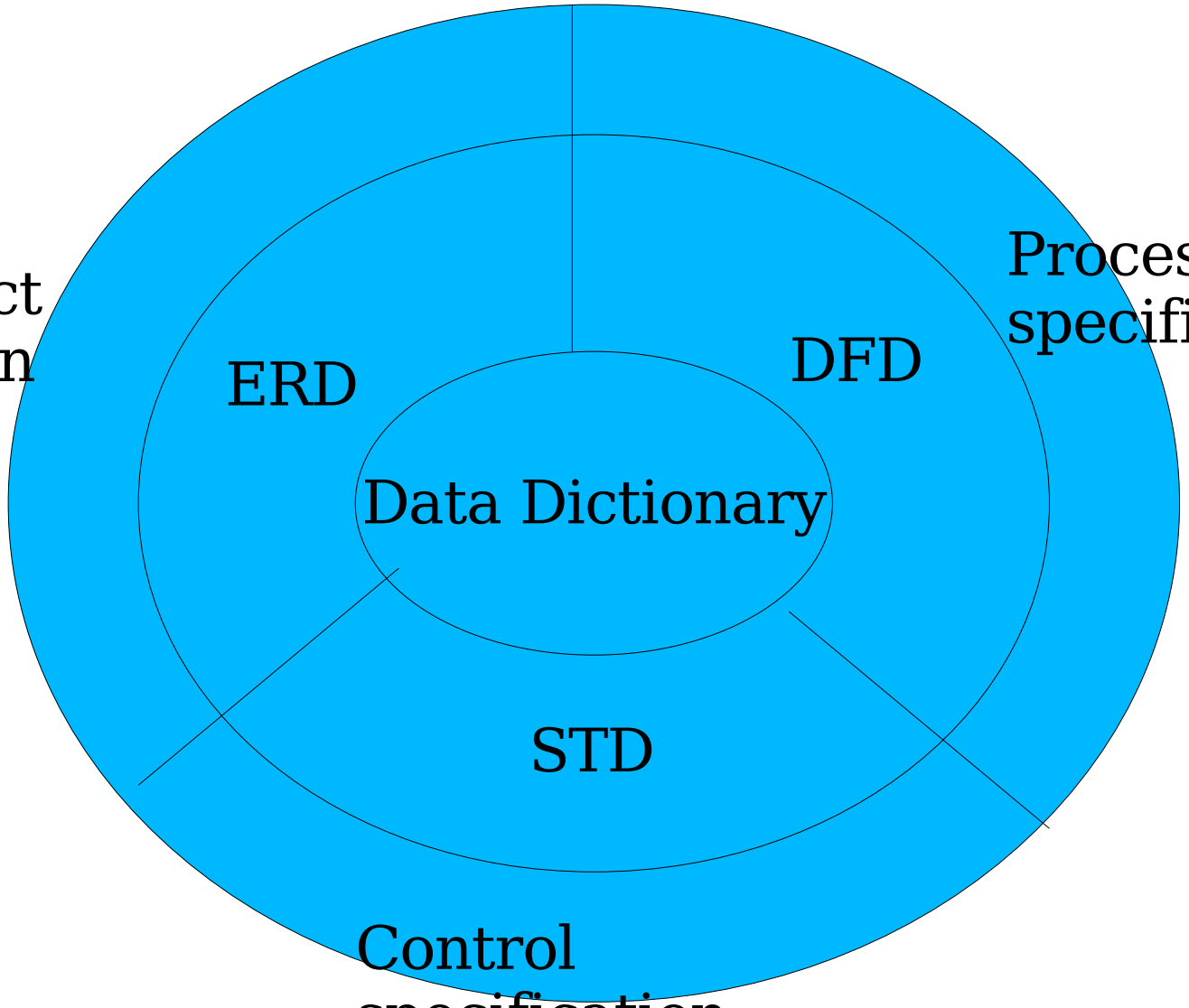
- Classical approach to analysis modeling
- Using notation that satisfies operational principles
  - Create models depicting information (data and control) content and flow
  - Partition the system functionally and behaviorally
  - Depict the essence of the system to be built

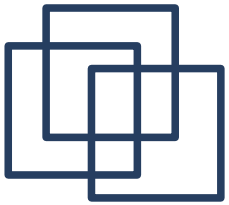


Analysis Model Elements

- Three primary objectives
  1) Describe what the customer requires
  2) Establish a basis for the creation of software design
  3) Define a set of requirements that can be validated once the software is built

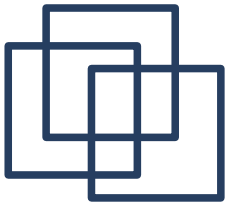
Structure of Analysis Model





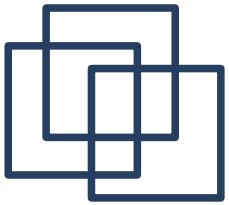
Analysis Model Elements

- Data Dictionary
  - A repository that contains descriptions of all data objects consumed or produced by the software
- Entity-Relationship Diagram (ERD)
  - Depicts relationships between data objects
  - Used to conduct the data modeling activity
  - Decribed using *data object description*



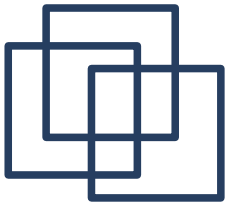
Analysis Model Elements

- Data Flow Diagram (DFD)
  - Provides an indication of how data are transmitted as they move through the system
  - Depict the functions (and sub-functions) that transform the data flow
  - Contained in the *Process Specification*
- State Transition Diagram (STD)
  - Indicates how the system behaves as a consequence of external events



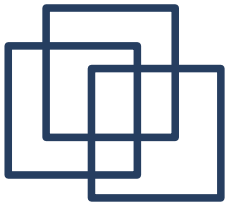
Analysis Model Elements

- State Transition Diagram (STD)
  - Represents the various modes of behavior called *states*
  - Manner in which transitions are made from state to state
  - Contained in the *Control Specification*



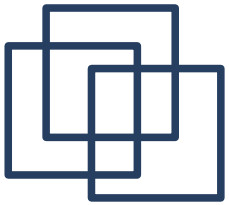
Data Modeling

- Uses ERD
  - Focuses solely on data
- Data Object
  - Representation of almost any *composite* information that must be understood by software
  - External entity, a thing, an occurrence, event, role, organizational unit, place, structure
  - No reference on the operation performed, thus can be represented as table



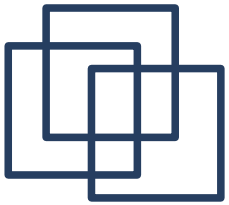
Data Modeling

- Attributes
  - Define the properties of a data object
    1)Name an instance of the data object
    2)Describe the instance
    3)Make reference to an instance in another table
  - One or more attributes must be defined as identifier or *key*
  - Set of attributes that is appropriate for a given data object depends on the problem context



Data Modeling

- Relationships
  - Connections between data objects (object pairs)
  - Nature of relationship depends on the context of the problem
- Cardinality
  - Represents the number of occurrences of objects in a given relationship
  - Maximum number of objects that can participate in a relationship
    - One-one, one-many, many-one, many-many

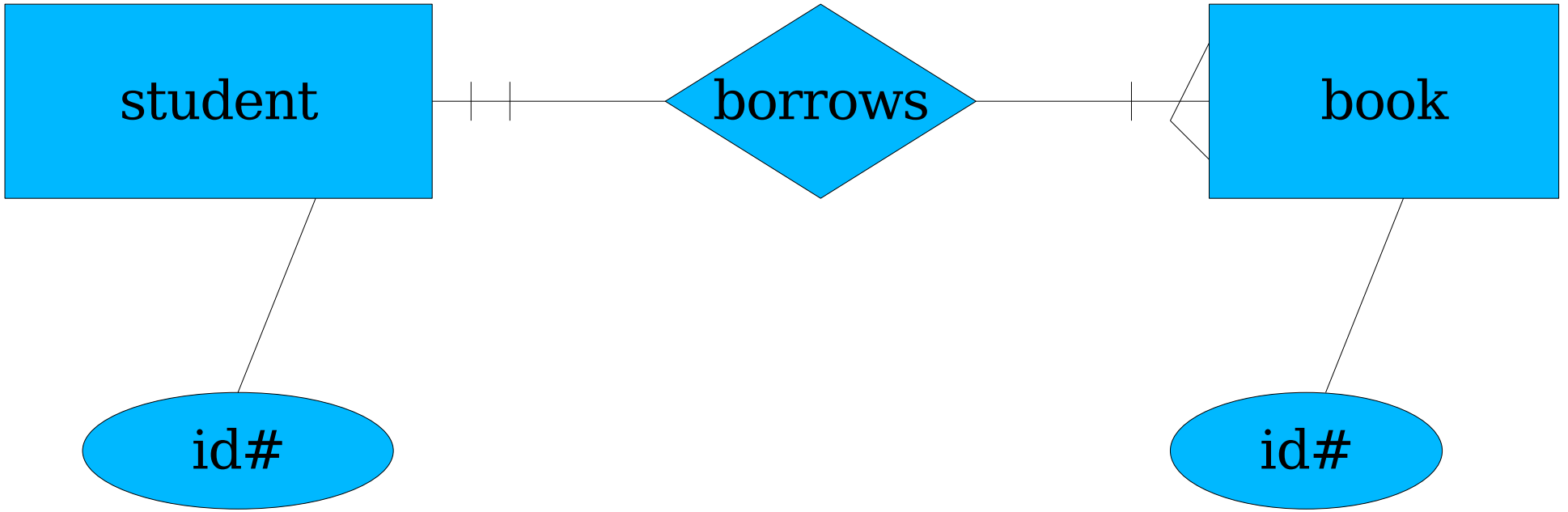


Data Modeling

- Modality
  - Zero if there is no explicit need for the relationship to occur, optional
  - One if the occurrence of a relationship is mandatory

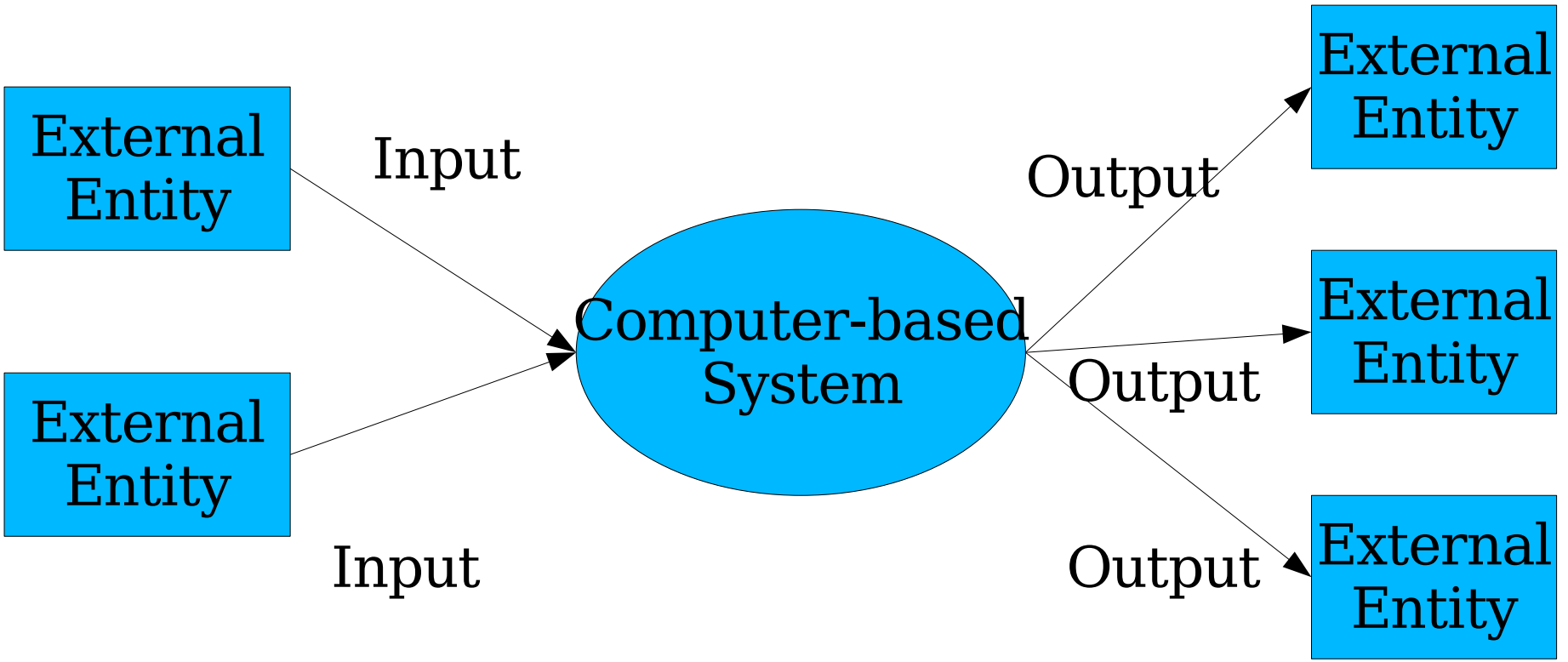
ERD

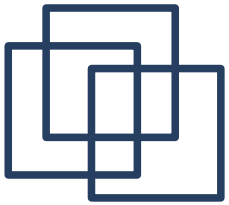
- Developed by Peter Chen (1977)



Functional Modeling and Information Flow

- Information is *transformed* as it flows through a computer-based system





DFD

external entity

Producer/consumer of information that resides outside system bounds

process

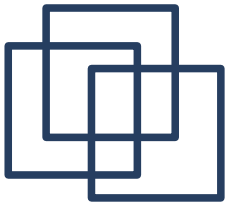
Transformer of information (function) that is within system bounds

data object

Data object, arrowhead indicates direction of data flow

data store

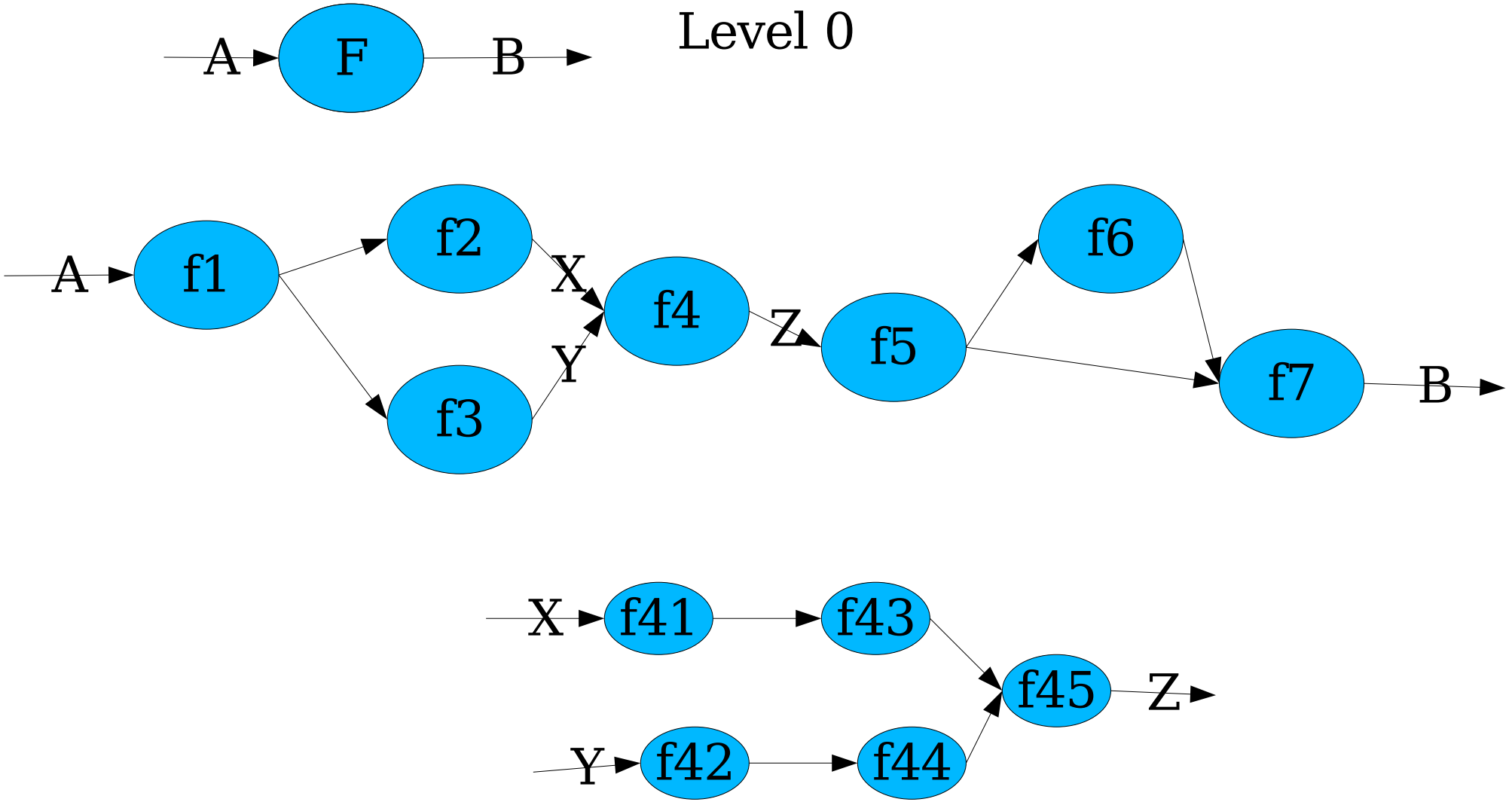
Repository of data, buffer or relational database

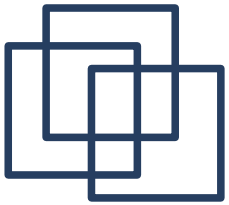


DFD

- Can be used to represent a system at any level of abstraction
- Level 0 DFD
  - Fundamental system model, context model
  - Represents system as single bubble with input and output
  - Further refinements can be made

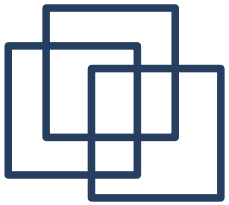
DFD





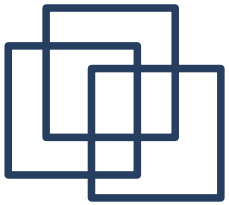
DFD

- Basic notation is not sufficient to describe requirements for software
  - What is the content of the data implied by arrow or data store?
    - Create a data dictionary
- Must be augmented with descriptive text: *Process Specification*
  - For each bubble: specify input, algorithm, output, limitations
- Extensions have been made for basic DFD



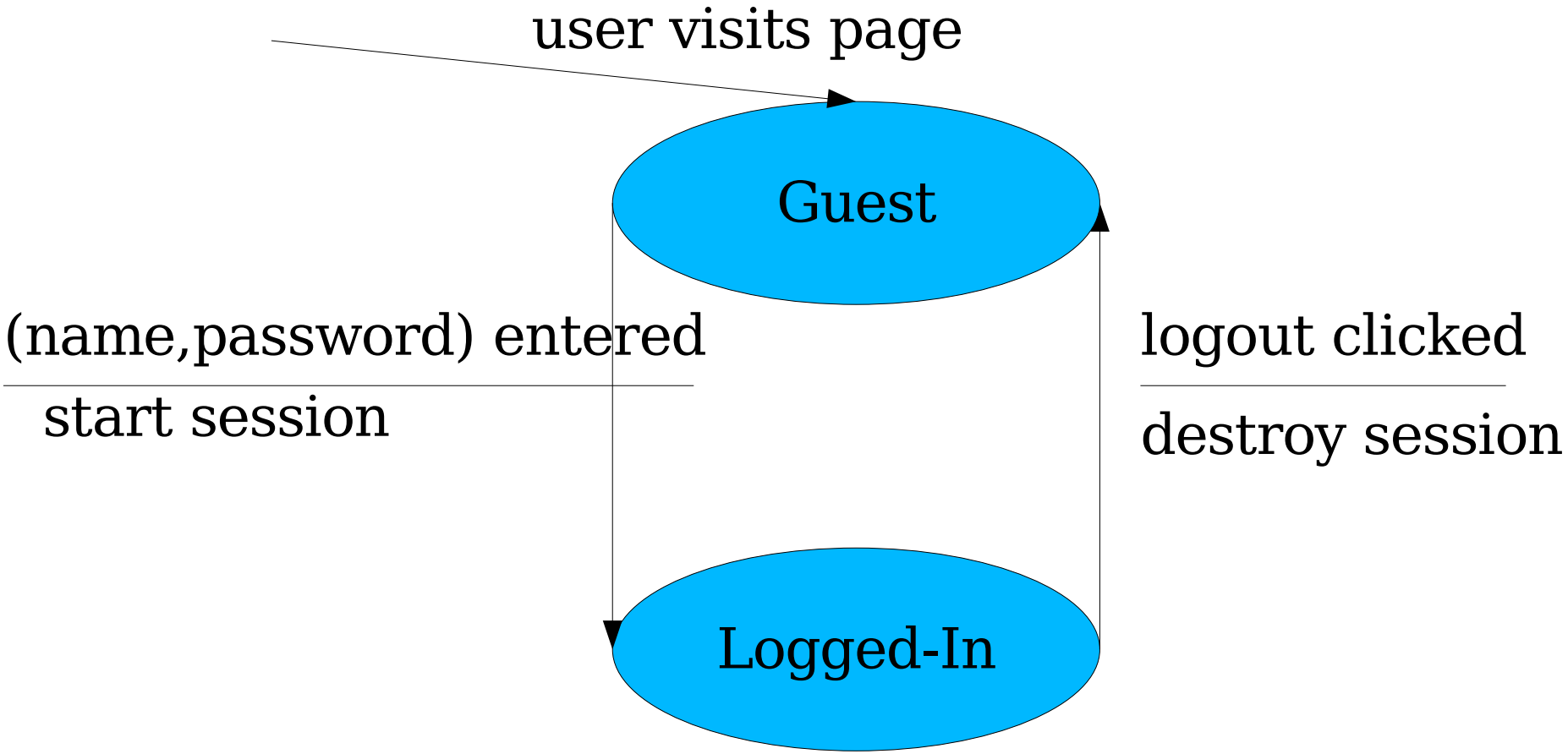
Behavioral Modeling

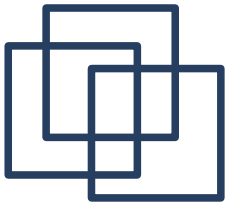
- Represented by State Transition Diagram (STD)
  - States and events that causes the change in state
  - Also indicated are actions to be taken when an event occurs
- *State* is any observable mode of behavior



STD

- Web Application example





Mechanics of SA

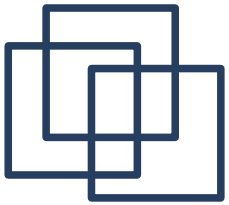
- Heuristics to derive good analysis model
- Creating an ERD

  1)Customers are asked to list the “things” that business process address

  2)Analyst determines if connections exists on the data objects

  3)When connection exists, create object-relationship pairs

  4)For each relationship pair, cardinality and modality are explored



Mechanics of SA

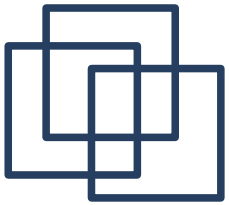
- Creating an ERD

  5)Steps 2 to 4 are continued iteratively until all pairs have been defined

  6)Attributes of each entity are defined

  7)An ERD is formalized and reviewed

  8)Steps 1 to 7 are repeated until data modeling is complete



Mechanics of SA

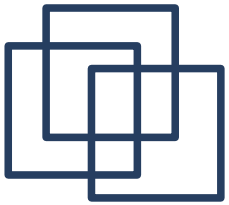
- Creating a DFD

  1)Successive refinements perform implicit functional decomposition of the system

  2)Level 0 DFD should depict system as a single bubble

  3)Primary input and output should be carefully noted

  4)Refinement should begin by isolating candidate process, data objects, and stores to be represented at the next level



Mechanics of SA

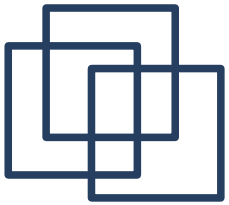
- Creating a DFD

  5)All arrows and bubbles should be labeled with meaningful names

  6)Information flow continuity must be maintained from level to level

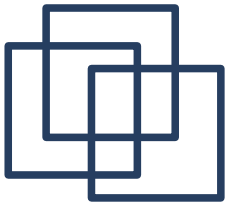
  7)One bubble at a time should be refined

  - Perform a grammatical parse of description: identify verbs (processes) and nouns (external entities, data/control objects, data stores)



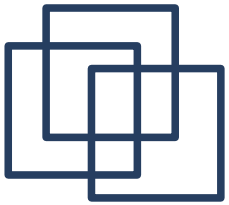
Process Specification

- Describes all flow model process that appear at the final level of refinement
- Include narrative text, program design language, math equations, tables, diagrams or charts
- Example: Consider a web application that has undergone refinement and the final DFD shows a 'check user process'.



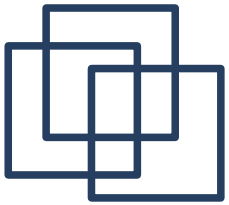
Data Dictionary

- Provides an organized approach for representing characteristics of each data object and control item
  - Name
  - Alias
  - Where-used, how-used
  - Content description
  - Supplementary information



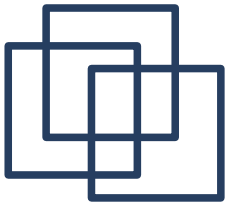
Summary

- Structured analysis relies on data modeling and flow modeling to create basis of comprehensive analysis model
- ERD represents the data objects and relationships in a system
- DFD is used to show the transformation of data and control
- Behavioral model is created using state transition diagram



Summary

- Data content is represented by a data dictionary
- Originally, structured analysis was used for data processing applications, but extensions are now available
- Structured analysis is supported by a variety of CASE tools



Reference

- Roger S. Pressman.Software Engineering: A Practitioner's Approach, 4th Ed.McGraw-Hill,1997. Chapter 12