

CS3773 Software Engineering

Lecture 11 Modeling Notations

Modeling Notations

Characteristics of a good modeling notation

- Well-defined set of concepts
- CASE tools support
- Resulting in unambiguous, clear, consistent, and concise specification
- Stakeholders understanding them (graphical notation is good)
- Ability to analyze (syntax, consistency checking)

2

UTSA CS3773

Basic Specification Notations

- Most specification languages are a combination of notations used to describe different aspects of system behavior
- Many languages contain variants of the following basic notations:
 - Entity-relation diagrams (ER Diagrams)
 - Event Traces
 - State machines
 - Data flow diagrams
 - Logic and functions
 - Algebraic specifications

3

UTSA CS3773

Decomposition Strategies

- Functional decomposition: break down a system into functions, which complete certain tasks
e.g., data flow diagram
- Process decomposition: break down a system into processes that can run concurrently in reaction to events
e.g., state machine
- Object Oriented decomposition: break down a system into objects and describe features of each object and their interactions
e.g., ER diagram

4

UTSA CS3773

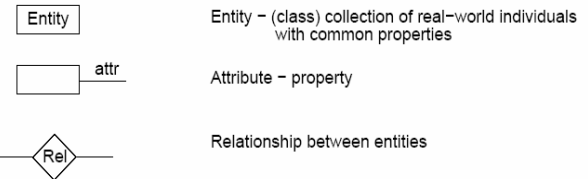
ER Diagram

- > ER diagram = entity-relation diagram
- > ER diagrams are used for database design
- > ER diagrams graphically represent problem domain:
 - Entities ("objects")
 - Attributes ("properties")
 - Relationships (often named with verbs)
- > In OO terms, the problem domain is called the conceptual level of description
 - Origins of class diagrams (UML)

5

UTSA CS3773

ER Diagram - Legend

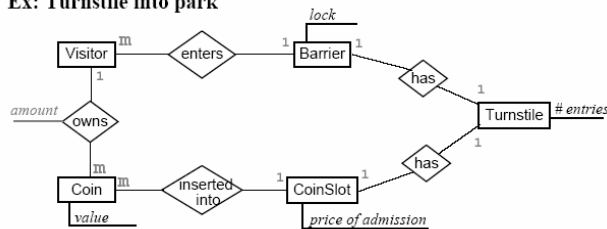


6

UTSA CS3773

ER Diagram - Example

Ex: Turnstile into park



7

UTSA CS3773

ER Diagram

- > **Advantages**
 - It is simple (few symbols)
 - It provides an overview of the system
- > **Disadvantages**
 - What are modeled as entities or attributes?
 - May encourage too much detail

8

UTSA CS3773

Event Traces

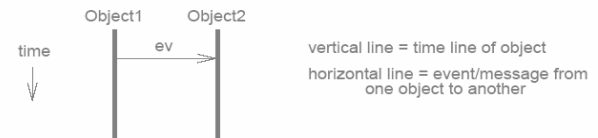
- Event traces graphically describe events/communication in one scenario
- Show messages between system and its environment (or among system entities)
- Represent iterations
- Assume asynchronous communication (non-blocking)

9

UTSA CS3773

Event Trace - Legend

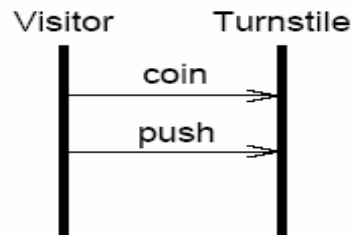
Legend:



10

UTSA CS3773

Event Trace - Example



11

UTSA CS3773

Event Trace

- Advantages:
 - Simple (one scenario)
 - Easy to understand
 - Somewhat precise (although timing isn't clear)
 - Useful for elicitation and consideration of end-to-end system behavior
- Disadvantages:
 - Not an efficient way to represent behavior

12

UTSA CS3773

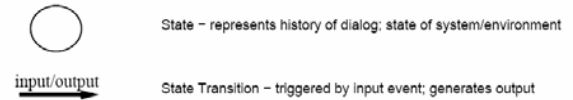
State Machine

- State machine describes dialog between system and environment graphically
- State machine shows control flow:
 - Behavior depends on current state
 - State represents the history of input
- State machine compactly represents of a set of event traces

13

UTSA CS3773

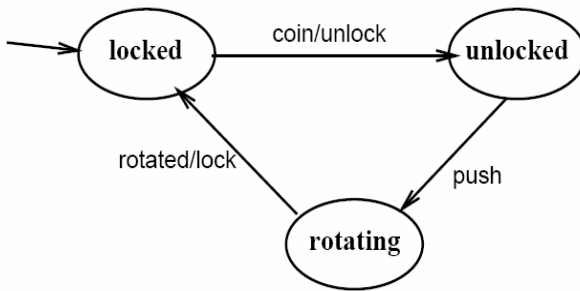
State Machine - Legend



14

UTSA CS3773

State Machine - Example



15

UTSA CS3773

State Machine

- Advantages
 - Most formal (least ambiguity)
 - Compact way to represent many behaviors
- Disadvantages
 - More complex
 - Not showing functional decomposition

16

UTSA CS3773

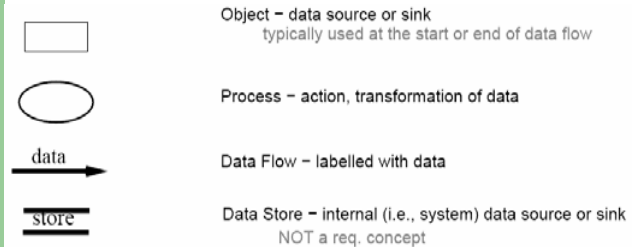
Data Flow Diagram

- > Data flow diagram (DFD) models high-level functionality of a system
- > DFD represents the flow of data among components graphically

17

UTSA CS3773

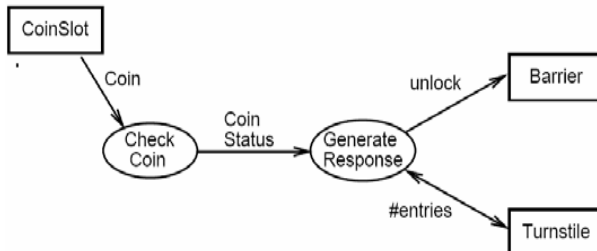
Data Flow Diagram - Legend



18

UTSA CS3773

Data Flow Diagram - Example



19

UTSA CS3773

Data Flow Diagram

- > **Advantages**
 - Simple
 - Good for functional decomposition
- > **Disadvantages — ambiguities**
 - When do functions get executed
 - If multiple inputs, are they all needed
 - Can't distinguish data and control signals
 - If multiple outputs, are both always generated

20

UTSA CS3773

Reading Assignments

- Sommerville's Book
 - Chapter 8, "System Models"
8.2, 8.3, 8.4