Object Behavior

- Behavior – actions that object carries out to provide requested service
- Description includes inputs, outputs, how object provides service
- Static behavior – depends only on inputs and attributes of object
- Dynamic behavior – depends on other objects

Understanding Control

- Identify all external and internal events to which system must respond
- For large systems, create use-case model for scenarios
- For each event, determine messages between objects and sequence
- Capture sequences using event-trace diagram

Static Behavior

- Approaches to specifying behavior
  - Stating pre- and post-conditions to execution
  - Decomposing service into series of tasks
- Distinguish between overall control of system (for use-case) and behavior of object

Example Event-Trace Diagram
Capturing Behavior

- Use scenarios to determine what each object service must do
- Record details of service behavior

Recording Conditions

- Pre-condition – what must be true before service provided
- Post-condition – what will be true after service
- Attributes modified
- Invariants of object – condition service must preserve

Example Conditions

Object: Circulation  
Service: checkout  
Input: book identity, patron identity  
Output: due date  
Precondition: catalog and patron list accessible  
Postcondition: if book in catalog and patron in patron list then circulation record added to circulation list  
Attributes modified: circulation list  
Invariant: catalog and patron list accessible

Recording Tasks

- Pseudo-code (Structured English)  
- Description of tasks using words with code-like structure (while, if-then, etc.)  
- Refer to attributes of objects  
- Best to augment with conditions – especially invariants

Example Pseudo-Code

Object: Circulation  
Service: checkout  
Input: book identity, patron identity  
Output: due date  
if (book in catalog and patron in patron list)  
    add circulation record (book, patron) to circ list;  
    return today + two weeks;  
else  
    handle error