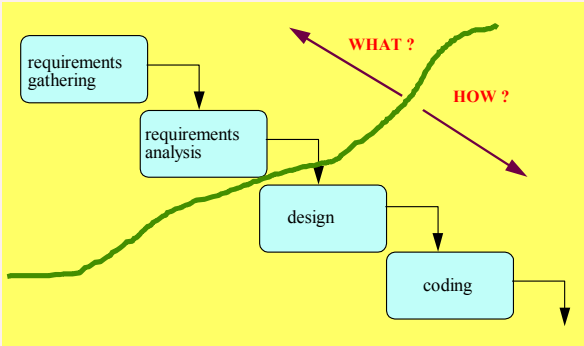
Traditional Development

Analysis / Design



Process models

- Usually Waterfall derivatives
- Data and Function based models
- Dataflow / Entity-Relation / Structure Diagrams (DFD ERD Structure Chart)
- RDBMs, supporting CASE tools

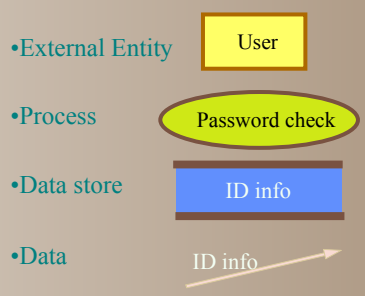
Common Methodologies

- Big organizations have their own
- Known methodologies:
  - Yourdon
  - Jackson
  - SSADM
  - SDM, SADT, DSSD ....

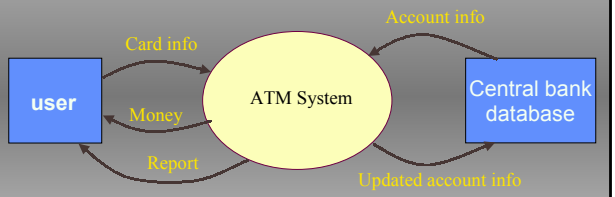
Common Approach

- Analysis:
  - DFD, ERD, Data Dictionary
- Design
  - DFD, Tables, Normalization, PSL/PDL
  - Data -> Structure -> Function
  - Transform Analysis
- 'Procedural' language + SQL

Analysis: DFD

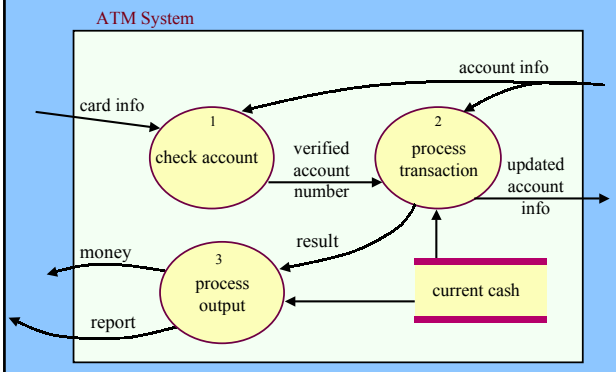


DFD: Context Diagram



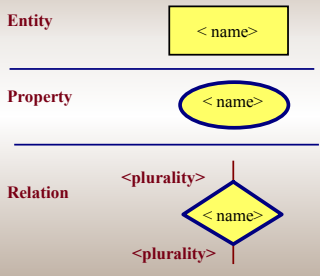
Ali Doğru METU 71

DFD: Level 1



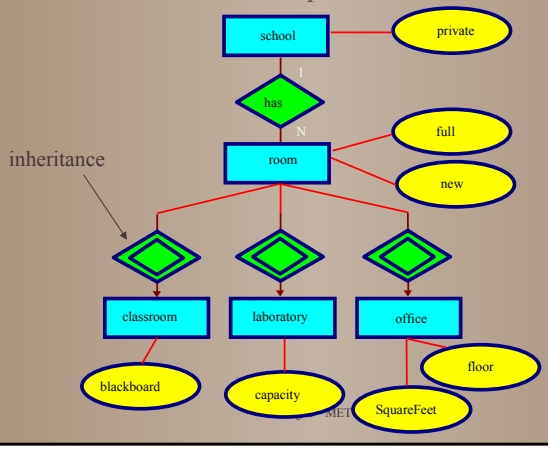
Ali Doğru METU 72

Entity Relationship Diagrams



Ali Doğru METU 73

An Example ERD



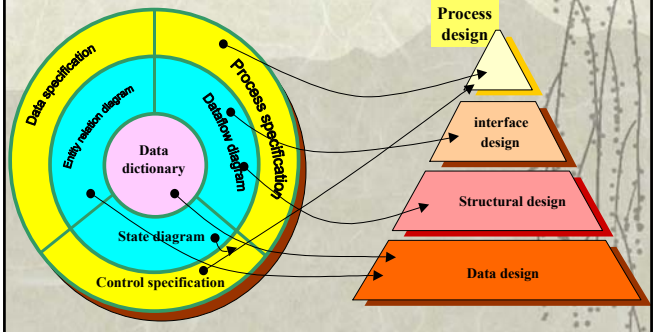
Ali Doğru METU 74

A Data Dictionary Example

Name	Description	Format	Date	Status
School	Name of the school	Text	9/25/01	cancelled
Floor	Attribute of the room entitiy: material	Text (hardwood, carpet)	9/26/01	Active
Capacity	Attribute of the laboratory entity	Integer	9/26/01	Active
Has	Relation: 1st entity owns the 2nd		9/26/01	active

Ali Doğru METU 75

Analysis to Design



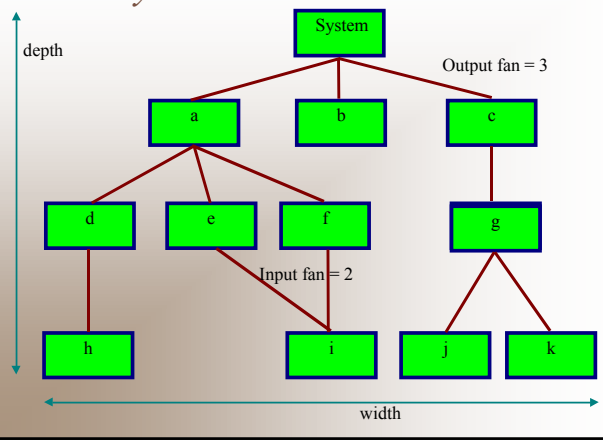
Ali Doğru METU 76

Modular Complexity

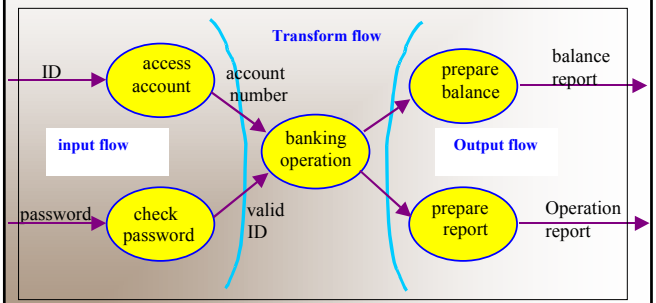
- Complexity:
- C(p1+p2) > C(p1) + C(p2)
- Functional Dependency:
  - Cohesion
  - Coupling

Ali Doğru METU 77

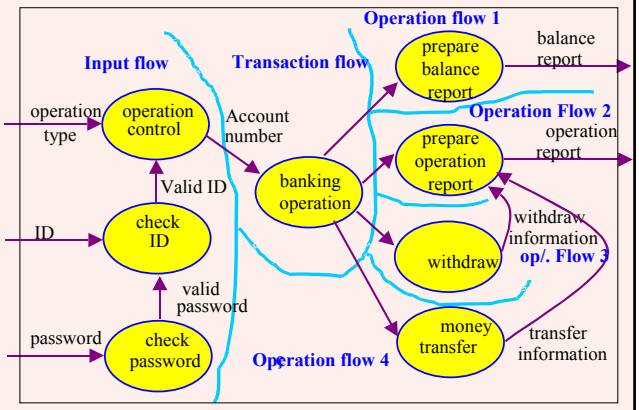
A System and its Modules



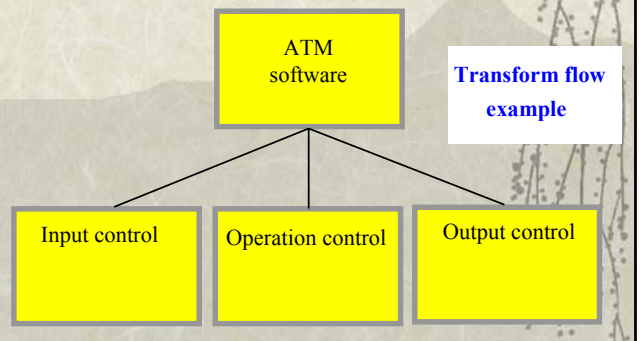
Transform: Transform Flow



Transaction Flow

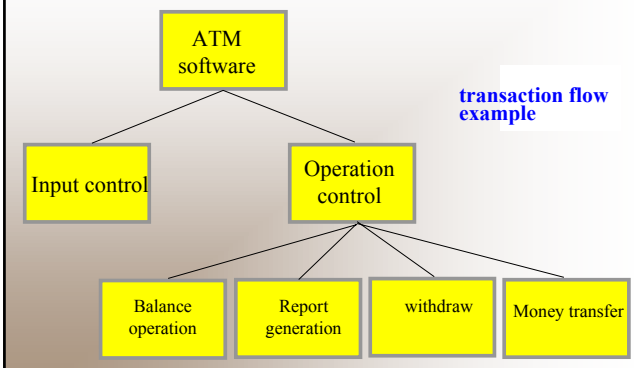


First Step in Transform



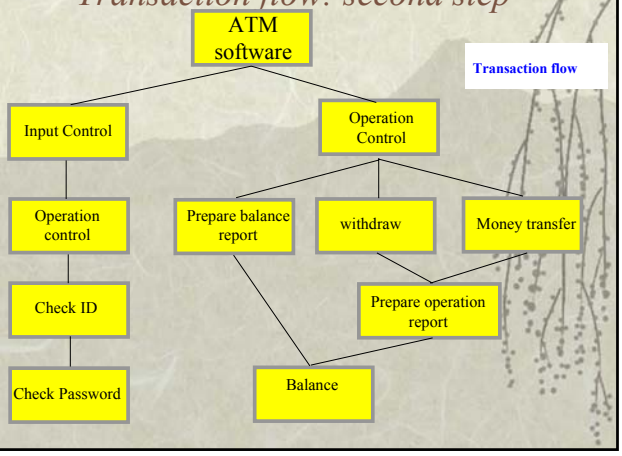
Ali Doğru METU 81

Transaction flow: first step



Ali Doğru METU 82

Transaction flow: second step



Procedural Specification: PDL

```
Do
    Read account number
    If (invalid account number) go to beginning
    Ask for operation type
    If (Operation type == deposit) then {Deposit(); go to beginning}
    Ask for password
    If (invalid password) go to beginning
    If (Operation type == balance) then {Balance(); go to beginning}
    If (insufficient balance) then go to beginning
    If (Operation type == money transfer) then {Transfer(); go to beginning}
    If (Operation type == money transfer) then {Transfer(); go to beginning}
    Else Withdraw()
While (continuous)
```


Testing

- Test Plan
- Test cases
- Testing techniques
- Testing strategies

- Unit Test
- Integration Test
- System Test
  - V&V
  - α & β

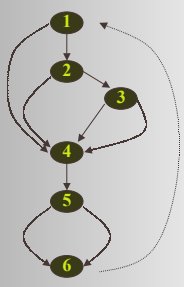
Testing Techniques

- Black box

- White box
  - Exhaustive testing: prohibitive complexity
  - Basis path testing: manageable:
    - Cyclomatic complexity =
      - # decision points + 1
      - N_n-N_e + 2
      - # Loops in a planar flow-graph

Basis path testing

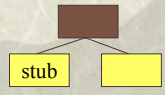
```
A = 2
While B <> A do      {1}
   If C > A          {2}
      D = C
   Else if D> C      {3}
      D = A
End while            {4}
If D > 2             {5}
   D = A
B = A                {6}
```



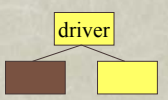
Cyc.Comp=9-6+2 = 5

Testing Strategies

- Top-down



- Bottom-up



- Sandwich

Object Oriented Approaches

- Programming languages: raising abstraction
- Languages in 80s, methodologies later
- A modeling that is closer to human mind than to machines
- Very similar OO modeling languages
- Consistent development media from analysis to coding
- Meanwhile maturing of SE; incorporating non OO techniques

Object Oriented Approaches - II

- Entities should be modeled with their operations
- Supports reuse and modularity
- Complexity is attempted to be addressed by generalization and specialization mechanisms
- Details to come soon…

Newly Developing Concepts

- Component Technologies
- Design Patterns
- Architectural Frameworks
- No methodological support yet !
- Domain orientation