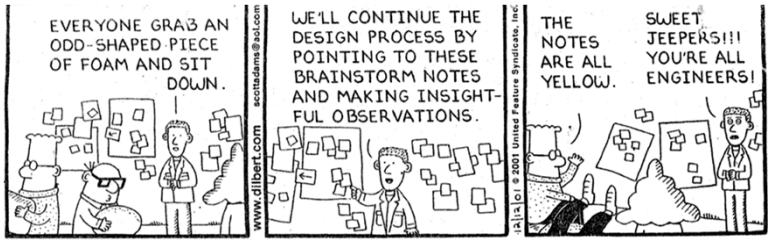
SEIS 610

Chapter 14

Agenda



- Design (Chapter 14)
  - Design and Abstraction (14.1)
  - Operation-Oriented Design (14.2)
  - Data Flow Analysis (14.3)
  - Transaction Analysis (14.4)
  - Data Oriented Design (14.5)
  - Object Oriented Design (14.6)
  - Test Workflow (14.10)
  - Real-time Design (14.13)

Chapter 14 Design

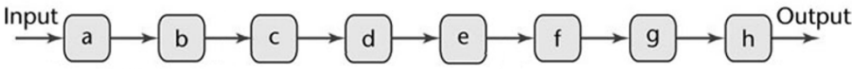
- **Architectural design** (14.1)
  - Modular decomposition of the product is our goal
  - We start with what we guess is the foundation
  - We build our (detailed) design on top!
- **Operation-Oriented** Design (14.2)
  - Remember high cohesion and low coupling
  - Books two approaches Data Flow Analysis and Transaction Analysis

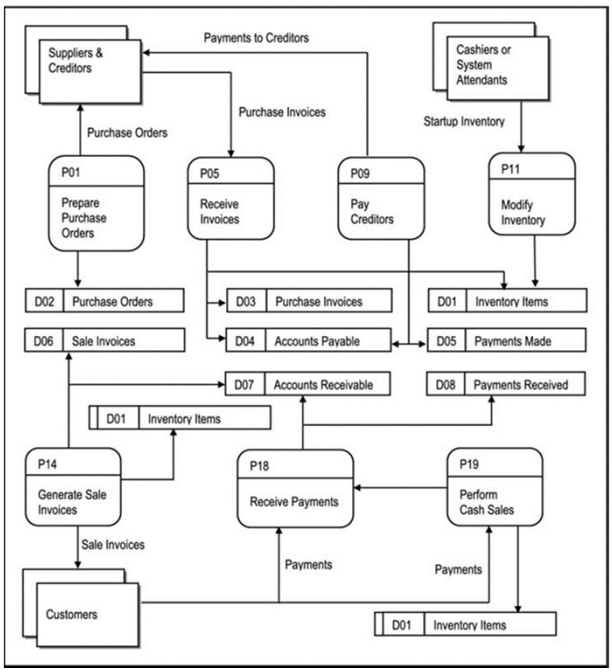
Data Flow Analysis (14.3)

- When DFD is done
- We have all the information we need
- Determine the point of highest abstraction
  - For input
  - For output
- Basically, where input looses the quality of being input and is only internal data.

Data Flow Analysis

- Software transforms input into output
- There happens to be no data stores in this picture
- Picture assumes all input goes into one process
- Picture assumes all output goes out one process
- This is really not how we have thought of it





Example

http://apprize.info/usability/engineering/engineering.files/image054.jpg

This is more indicative of how we have thought of things.

Data Flow Analysis (14.3)

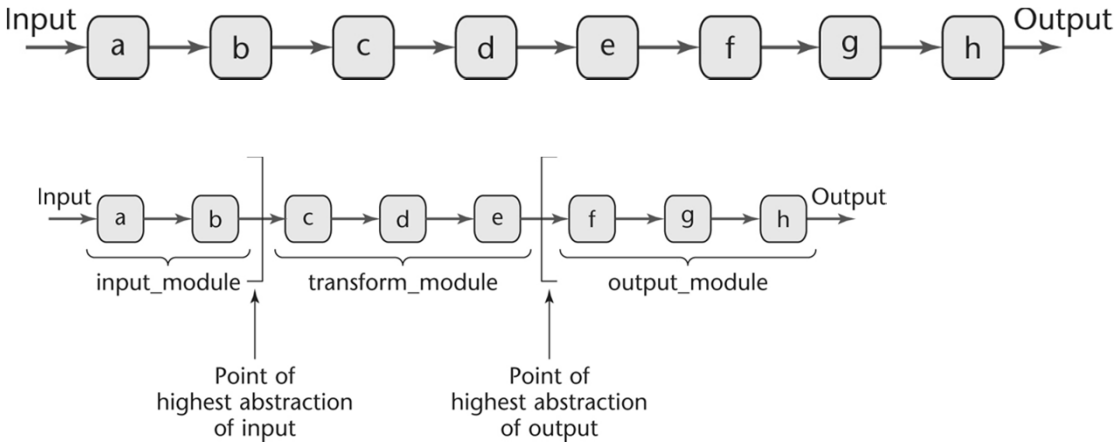
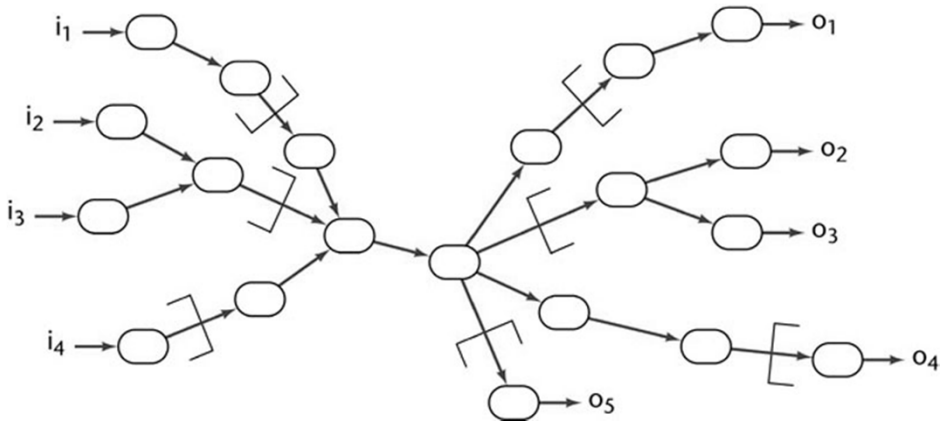
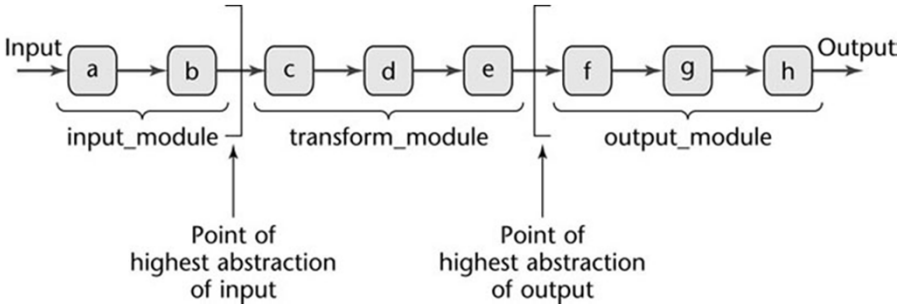


Figure 14.8



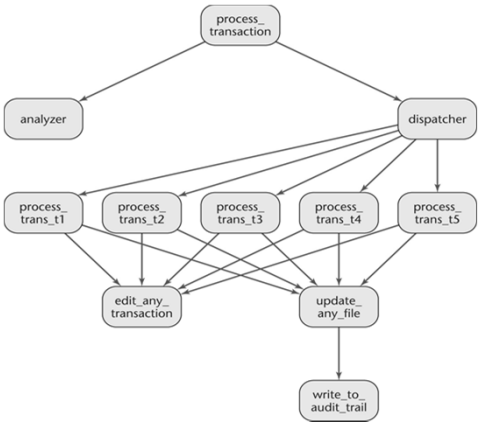
Data Flow Analysis

- Lather, rinse, repeat.
- Decompose product into modules
- Repeat until you believe you have high cohesion



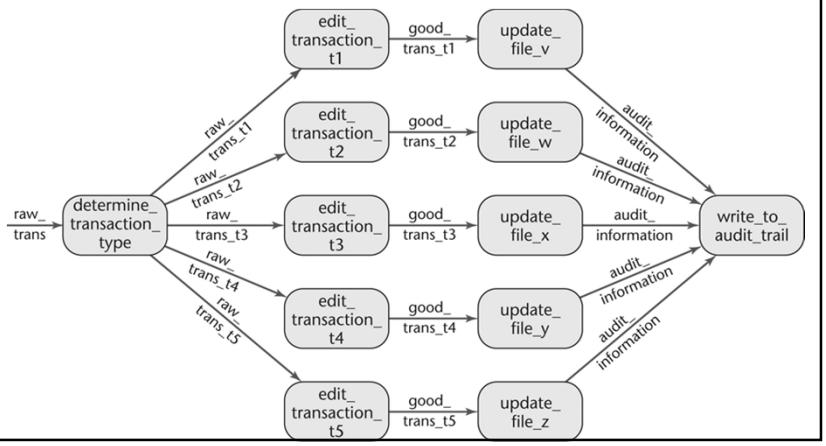
Transaction Analysis

- DFA might not work for transaction processing products
- This solution has 'control' coupling



Transaction Analysis

- Instead, perhaps have two handlers…One to edit and one to update.
- Instantiate as necessary.



Suppose we have a transaction

(jumping to Object Oriented)

- Transaction
  - Transaction amount
  - Transaction date

Suppose we have a transaction

- Transaction
- Data
  - Transaction amount (floating point number)
  - Transaction date (string)

Suppose we have a transaction class

- Transaction
- Data
  - Transaction amount (floating point number)
  - Transaction date (string)
- Methods
  - Set transaction amount
  - Edit transaction amount

Now think we have five different types of transactions!

- Loan transaction
- Savings transaction
- Sales transaction
- Return transaction
- Payment transaction

Vocab (you need to know this)

- Polymorphism
  - The condition of occurring in several different forms.
  - the occurrence of different forms among the members of a population or colony, or in the life cycle of an individual organism.
- Inheritance
  - Attributes from ancestor available to descendants
- Dynamic Binding
  - Correct object created when asked for!
- Information Hiding
  - Making only what is necessary available publicly
- Encapsulation
  - Bundling methods and data into objects. (or perhaps mechanisms and data)
- Abstraction
  - **F**ocus on the essential properties of a 'thing' instead of one specific example. (lynda.com)
  - We define essential aspects of a system.

Now suppose we have a file class

- Data
  - File name
  - File handle
- Methods
  - Update

Polymorphism

- Transaction
  - LoanTransaction
  - SavingsTransacction
  - SalesTransaction
  - ReturnTransaction
  - PaymentTransaction
- TransactionFile
  - LoanTransactionFile
  - SavingsTransactionFile
  - SalesTransactionFile
  - ReturnTransactionFile
  - PaymentTransactionFIle

So if we were to do this in Java

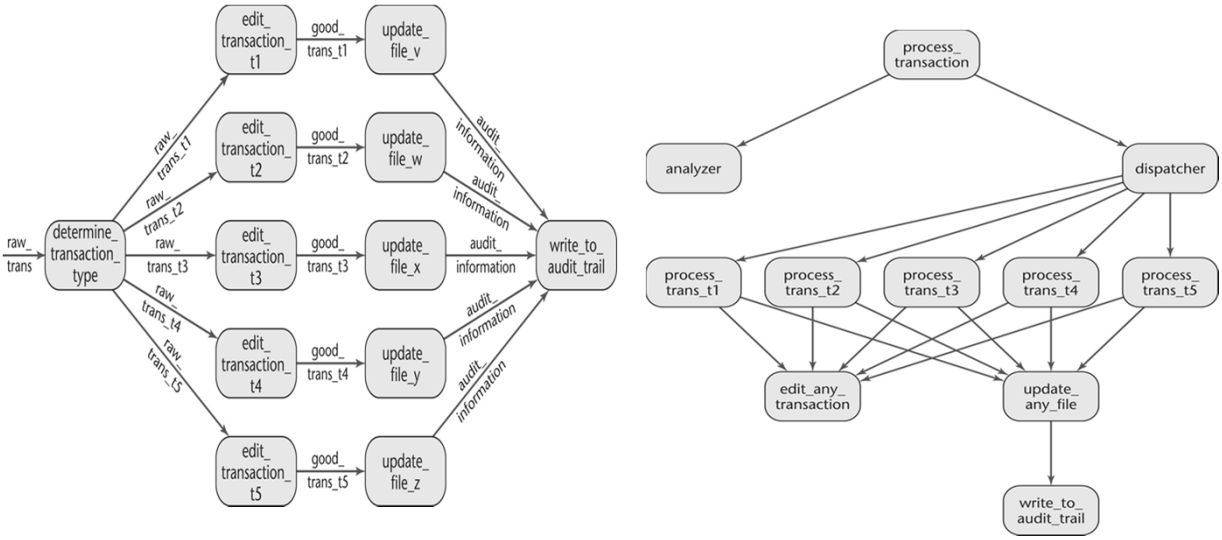
```
Transaction transaction = new Transaction();
File file = new TransactionFile();

transaction.edit(123.00);
file.update(transaction);
```

So if we were to do this in Java

```
Transaction transaction = new LoanTransaction();
File file = new LoanTransactionFile();

transaction.edit(123.00);
file.update(transaction);
```



Data-Oriented Design (14.5)

- Basic principle
  - The structure of a product must conform to the structure of its data
- Three very similar methods
  - Michael Jackson [1975], Warnier [1976], Orr [1981]
- Data-oriented design
  - Has never been as popular as action-oriented design
  - With the rise of OOD, data-oriented design has largely fallen out of fashion

14.6 Object Oriented Design

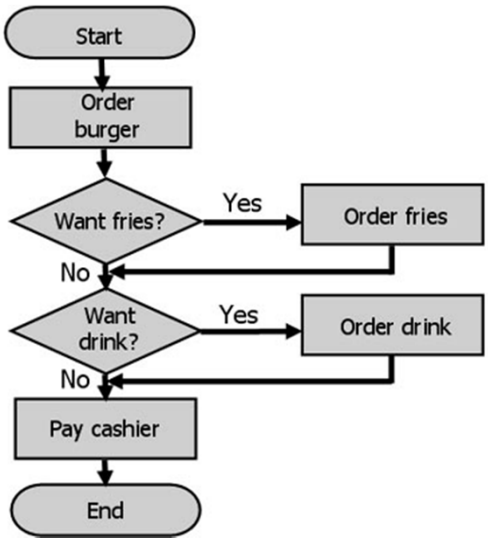
- Design product in terms of classes extracted during our analysis
  - Classes are identified during the object-oriented analysis workflow
  - Classes are designed during the design workflow
- SEIS 610
  - We write our play
  - We have our sequence diagrams
- Create a class diagram
  - With methods!
- Perform Detailed Design as required
  - State Diagrams
  - Activity Diagrams

Design Workflow (14.9)

- Complete the sequence diagram (dynamic)
- Complete the class diagram (static)
- Perform the detailed design
  - Dynamic diagrams
    - State charts
    - Flow charts
    - Petri Nets
    - DFD

Activity Diagrams

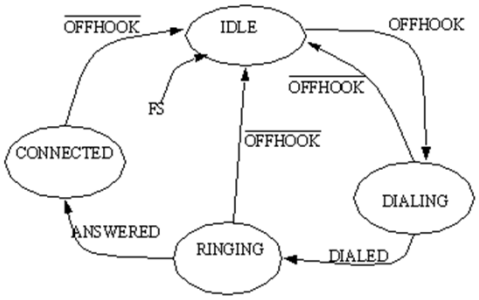
- "Old fashioned flowcharts"
- Use for methods with difficult logic.



- http://www.teach-ict.com/as_a2_ict_new/ocr/A2_G063/331_systems_cycle/analysis_tools/miniweb/images/flowchart.jpg

State Diagrams

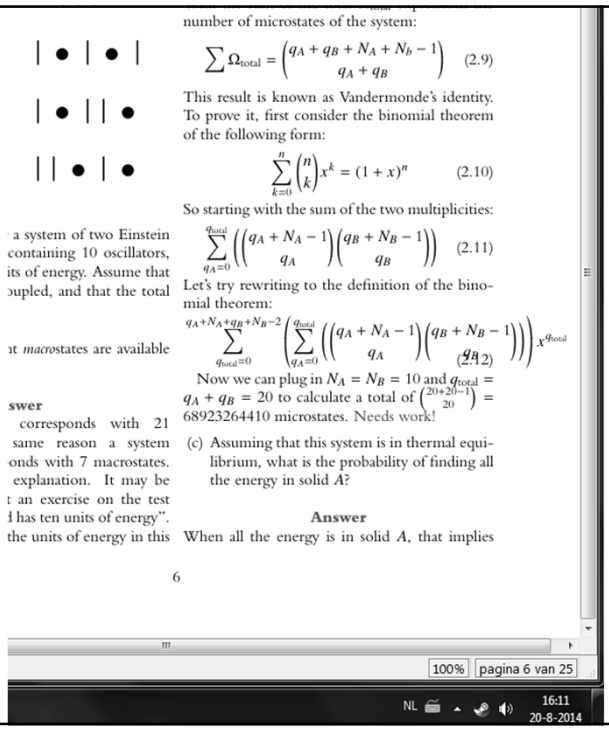
- Use for classes which have Interesting state logic.



http://www.thelearningpit.com/hj/plcs_files/plcs-353.gif

Text!

- Yes, you can use text!
- http://i.imgur.com/Bbn2CZj.png?1



| • | • |

| • | | • |

| | • | • |

a system of two Einstein containing 10 oscillators, its of energy. Assume that oupled, and that the total

ıt *macro*states are available

swer

corresponds with 21 same reason a system onds with 7 macrostates. explanation. It may be t an exercise on the test l has ten units of energy". the units of energy in this

number of microstates of the system:

$$\sum \Omega_{\text{total}} = \binom{q_A + q_B + N_A + N_b - 1}{q_A + q_B} \quad (2.9)$$

This result is known as Vandermonde's identity. To prove it, first consider the binomial theorem of the following form:

$$\sum_{k=0}^{n} \binom{n}{k} x^k = (1+x)^n \quad (2.10)$$

So starting with the sum of the two multiplicities:

$$\sum_{q_A=0}^{q_{\text{total}}} \left(\binom{q_A + N_A - 1}{q_A} \binom{q_B + N_B - 1}{q_B} \right) \quad (2.11)$$

Let's try rewriting to the definition of the binomial theorem:

$$\sum_{q_{\text{total}}=0}^{q_A+N_A+q_B+N_B-2} \left(\sum_{q_A=0}^{q_{\text{total}}} \left(\binom{q_A + N_A - 1}{q_A} \binom{q_B + N_B - 1}{q_B} \right) \right) x^{q_{\text{total}}} \quad (2.12)$$

Now we can plug in $N_A = N_B = 10$ and $q_{\text{total}} = q_A + q_B = 20$ to calculate a total of $\binom{20+20-1}{20} = 68923264410$ microstates. Needs work!

(c) Assuming that this system is in thermal equilibrium, what is the probability of finding all the energy in solid A?

Answer

When all the energy is in solid A, that implies

6

100% pagina 6 van 25

NL 16:11 20-8-2014

Do we have to do all diagram types?

- Nope!
- **You should always do sequence and class diagrams**
- You may need activity diagrams
- You may need state diagrams
- You may want Petri Nets
- You may like Data Flow Diagrams
- You may need a darn good description in text.
- But you don't need all of the above!

Keep in mind

- Information hiding
- Abstraction
- Responsibility-driven design
  - That is what we are doing!!

- The end!

- P.S. Forgot Test Workflow

Motto: Review Review Review