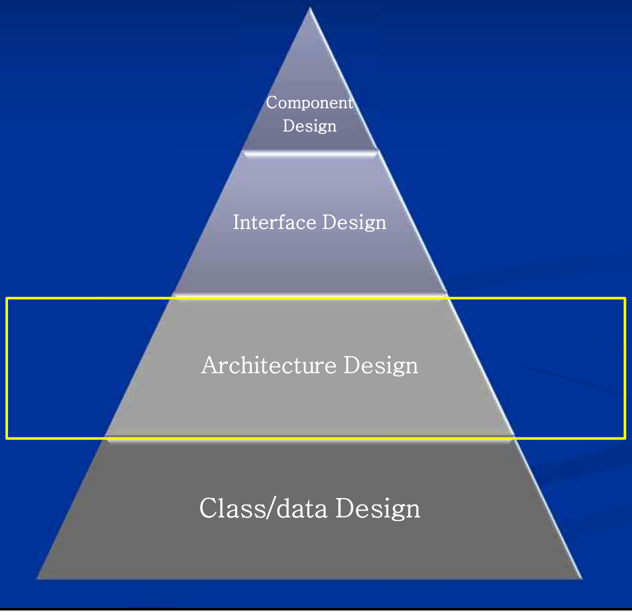
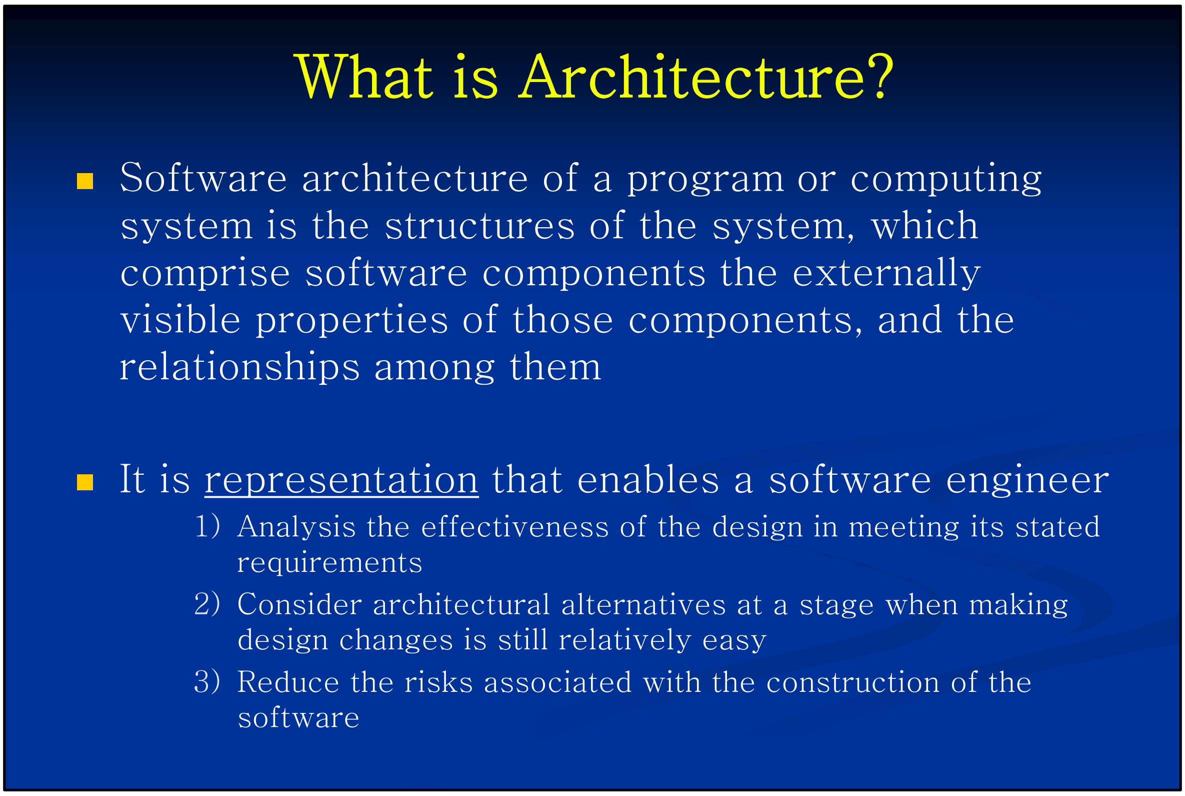
아키텍처 설계

한국항공대학교 소프트웨어학과 지승도교수
R.S. Pressman

아키텍처 설계

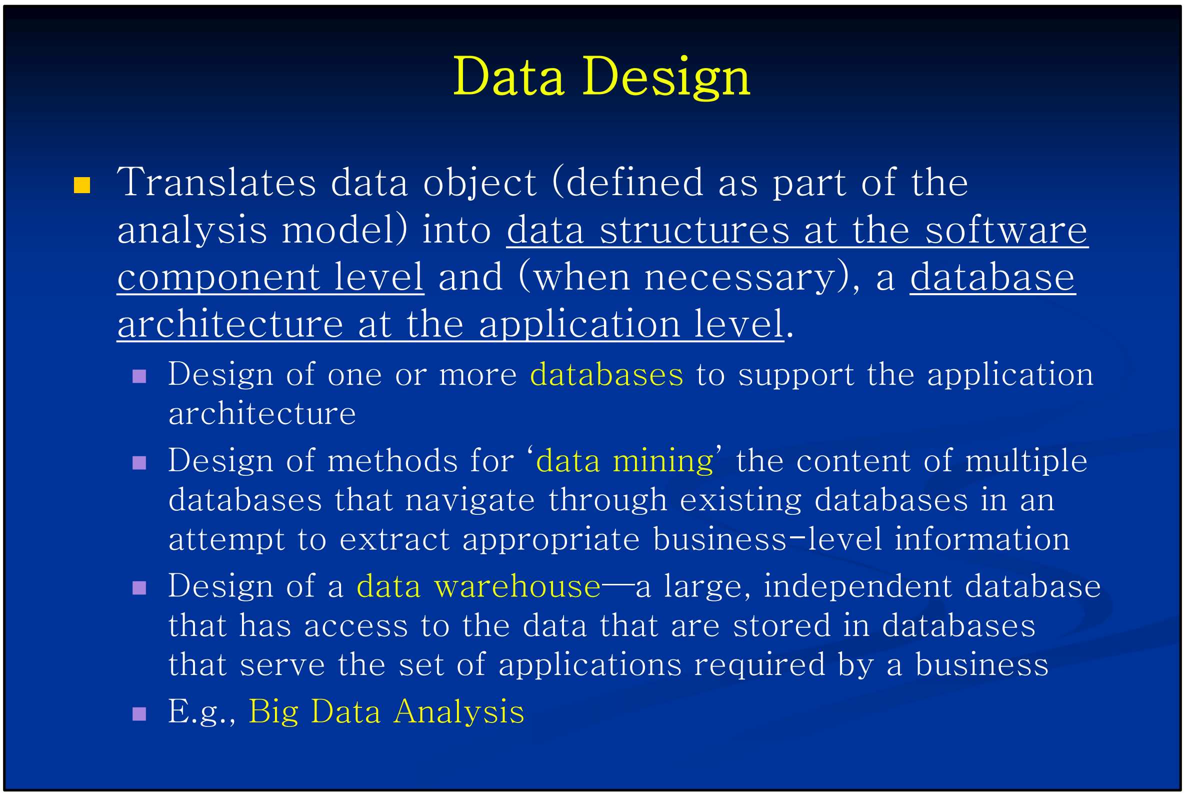




What is Architecture?

- Software architecture of a program or computing system is the structures of the system, which comprise software components the externally visible properties of those components, and the relationships among them

- It is representation that enables a software engineer
  1) Analysis the effectiveness of the design in meeting its stated requirements
  2) Consider architectural alternatives at a stage when making design changes is still relatively easy
  3) Reduce the risks associated with the construction of the software



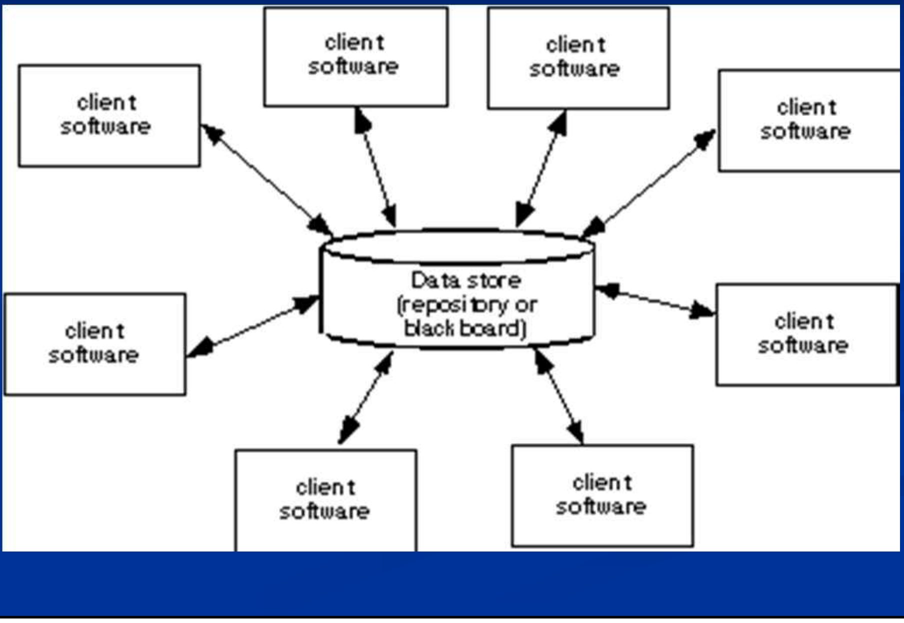
Data Design

- Translates data object (defined as part of the analysis model) into data structures at the software component level and (when necessary), a database architecture at the application level.
  - Design of one or more databases to support the application architecture
  - Design of methods for ‘data mining’ the content of multiple databases that navigate through existing databases in an attempt to extract appropriate business-level information
  - Design of a data warehouse—a large, independent database that has access to the data that are stored in databases that serve the set of applications required by a business
  - E.g., Big Data Analysis

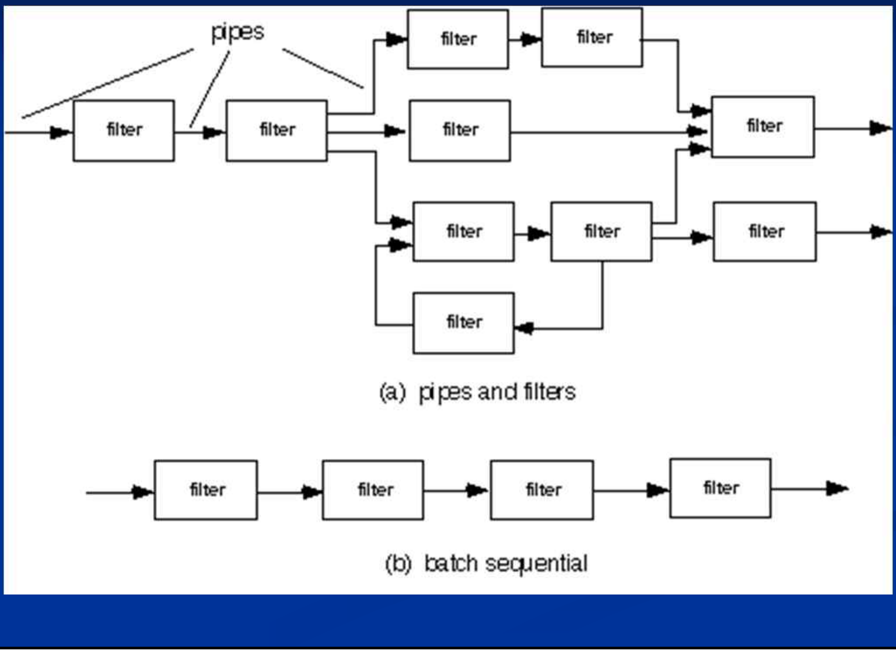
Architectural Styles

- Data-centered architectures
- Data flow architectures
- Call and return architectures
- Layered architectures

Data-Centered Architecture



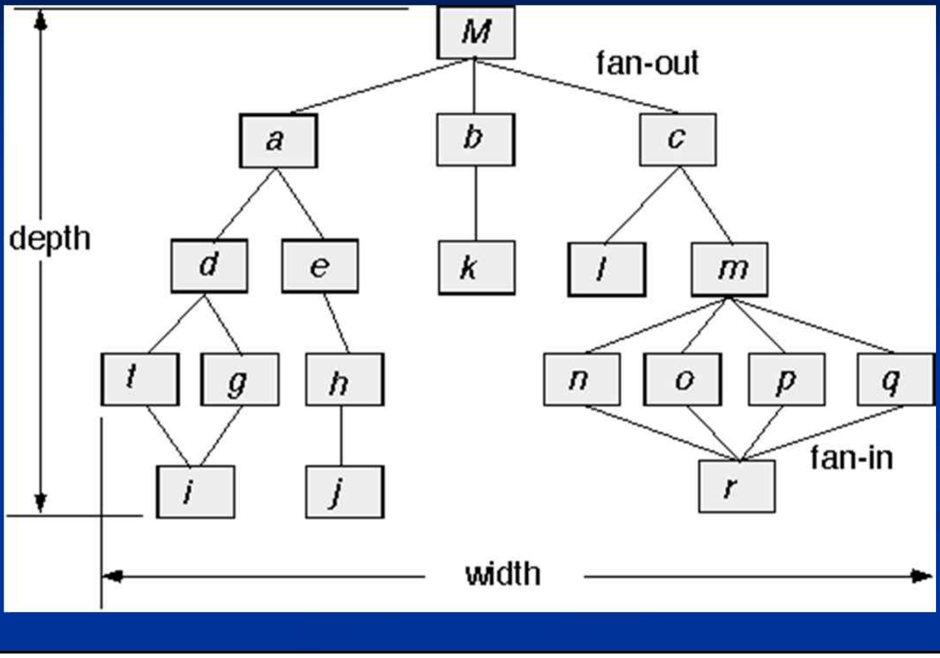
Data Flow Architecture



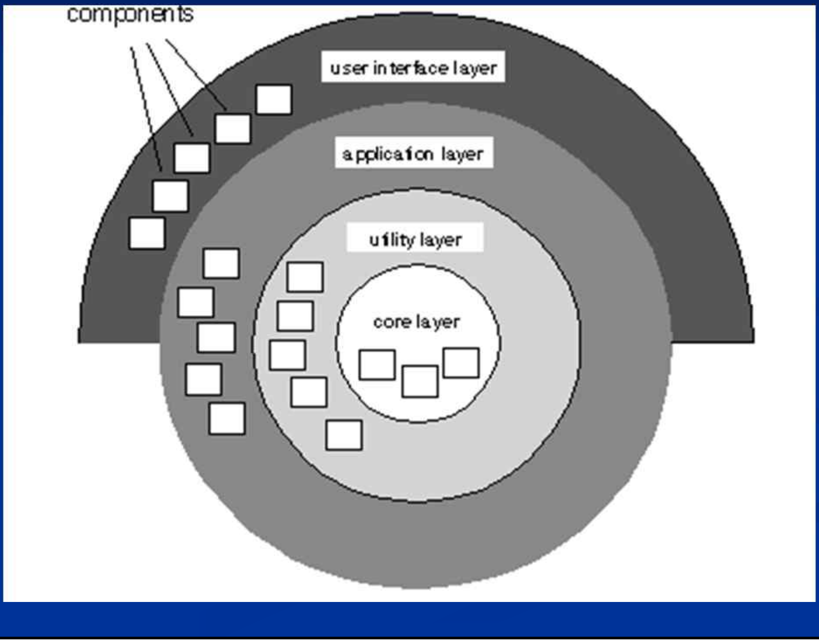
(a) pipes and filters

(b) batch sequential

Call and Return Architecture



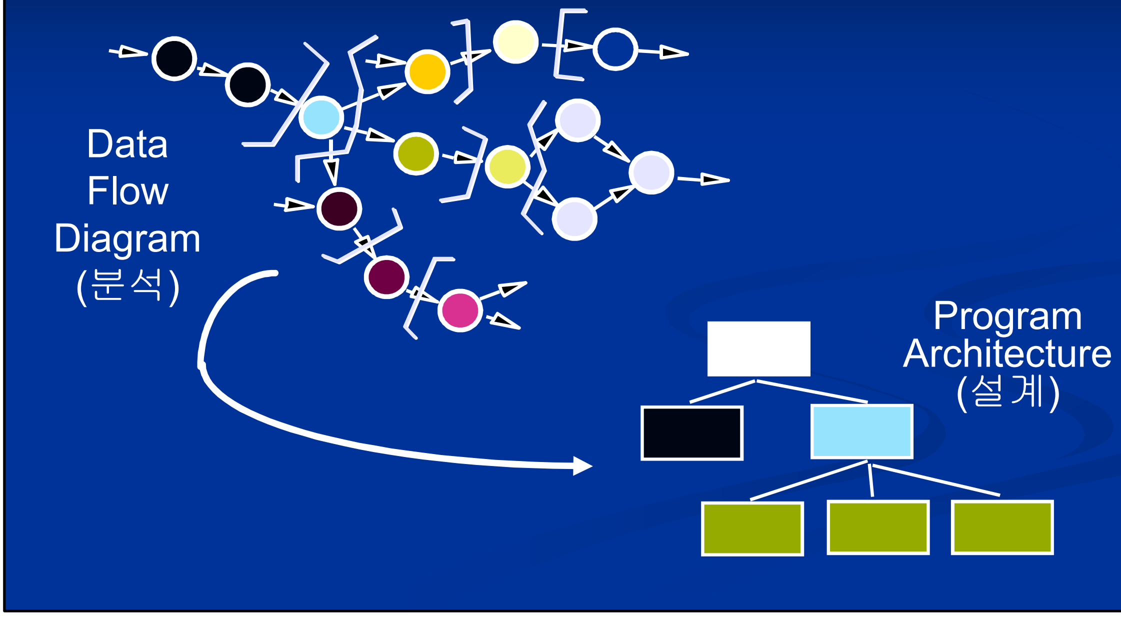
Layered Architecture



Mapping Data Flow (DFD) Into a Software Architecture

- Call and return architecture
- Transform Flow
  - Incoming flow
  - Transform center
  - Outgoing flow
- Transaction Flow
  - Transaction that triggers other data flow along one of many paths
  - Action paths
  - A transaction center

Deriving Program Architecture

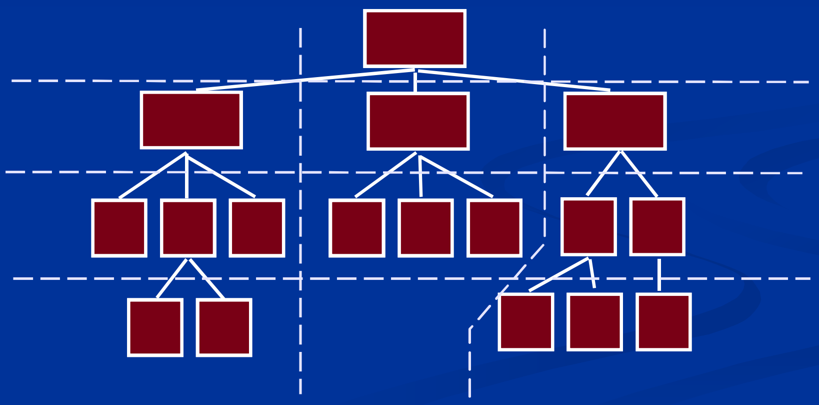


Data Flow Diagram (분석)

Program Architecture (설계)

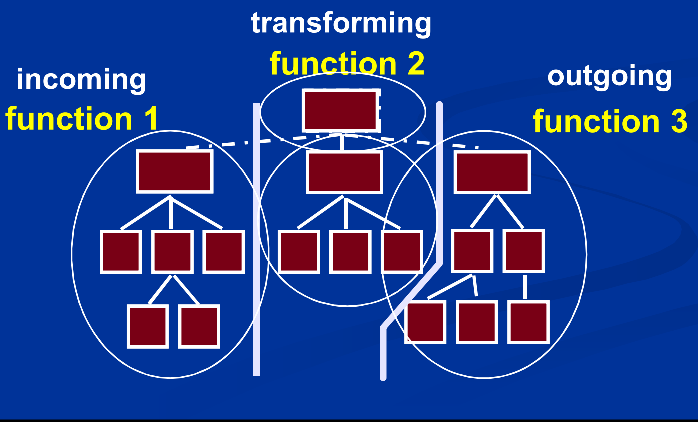
Partitioning the Architecture

- “horizontal” & “vertical” partitioning are required



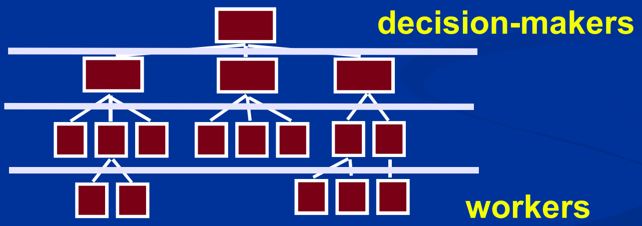
Horizontal Partitioning

- define separate branches of the module hierarchy for each major function
- use control modules to coordinate communication between functions

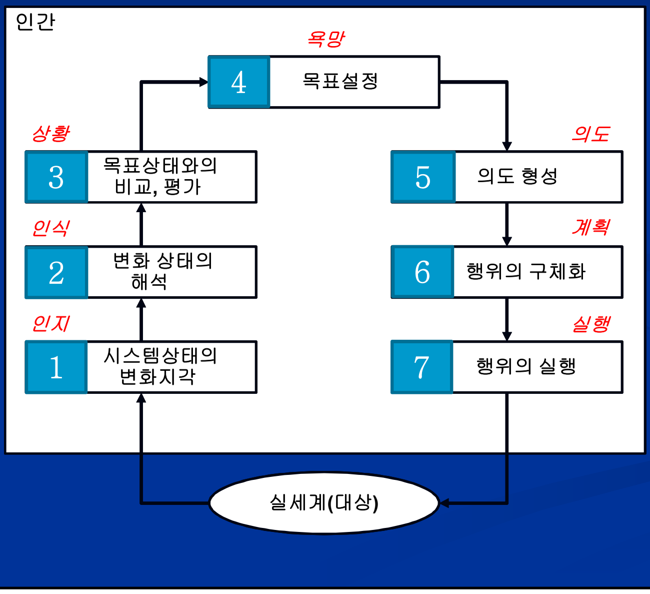


Vertical Partitioning: Factoring

- design so that decision making and work are stratified
- decision making modules should reside at the top of the architecture



7단계 인간 행위 모형 (by 인지심리학 Norman)



인간

욕망

4 목표설정

상황

3 목표상태와의 비교, 평가

인식

2 변화 상태의 해석

인지

1 시스템상태의 변화지각

의도

5 의도 형성

계획

6 행위의 구체화

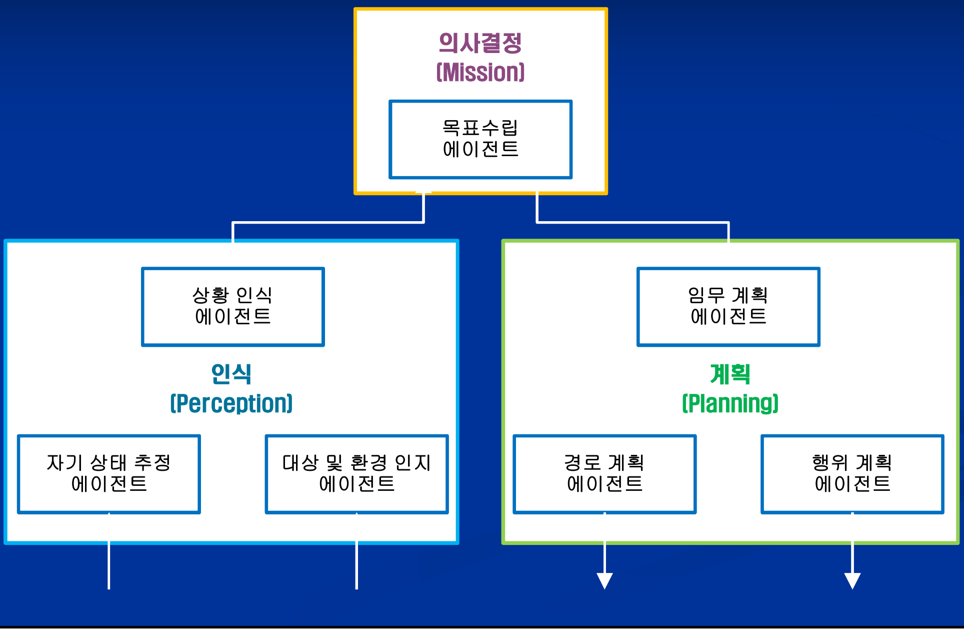
실행

7 행위의 실행

실세계(대상)

1. Perceiving World State
2. Interpreting World State
3. Evaluating Outcome
4. Forming the Goal
5. Forming the Intention
6. Specifying an Action
7. Executing an Action

AI 기반 시스템 구조



의사결정 (Mission)

목표수립 에이전트

상황 인식 에이전트

인식 (Perception)

자기 상태 추정 에이전트

대상 및 환경 인지 에이전트

임무 계획 에이전트

계획 (Planning)

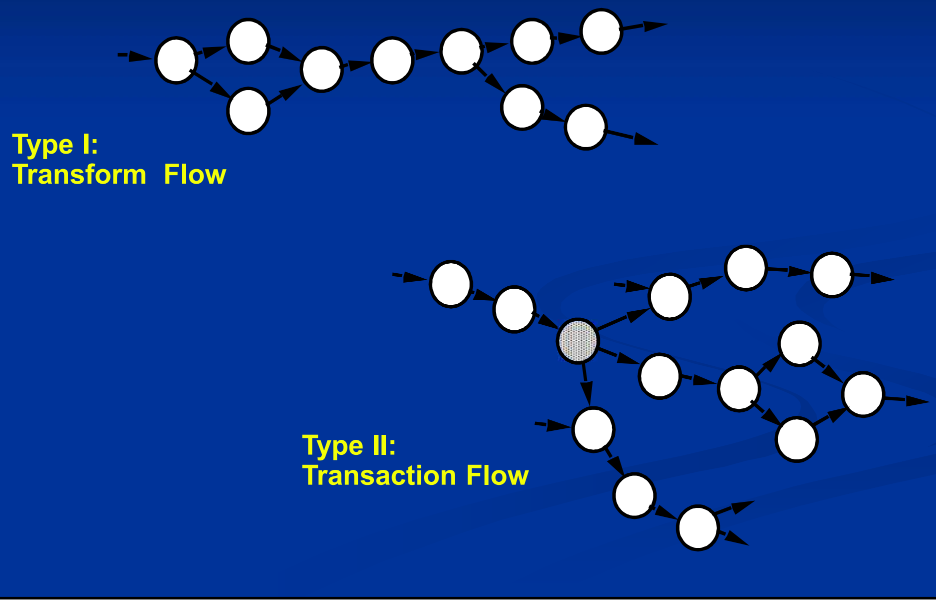
경로 계획 에이전트

행위 계획 에이전트

Why Partitioned Architecture?

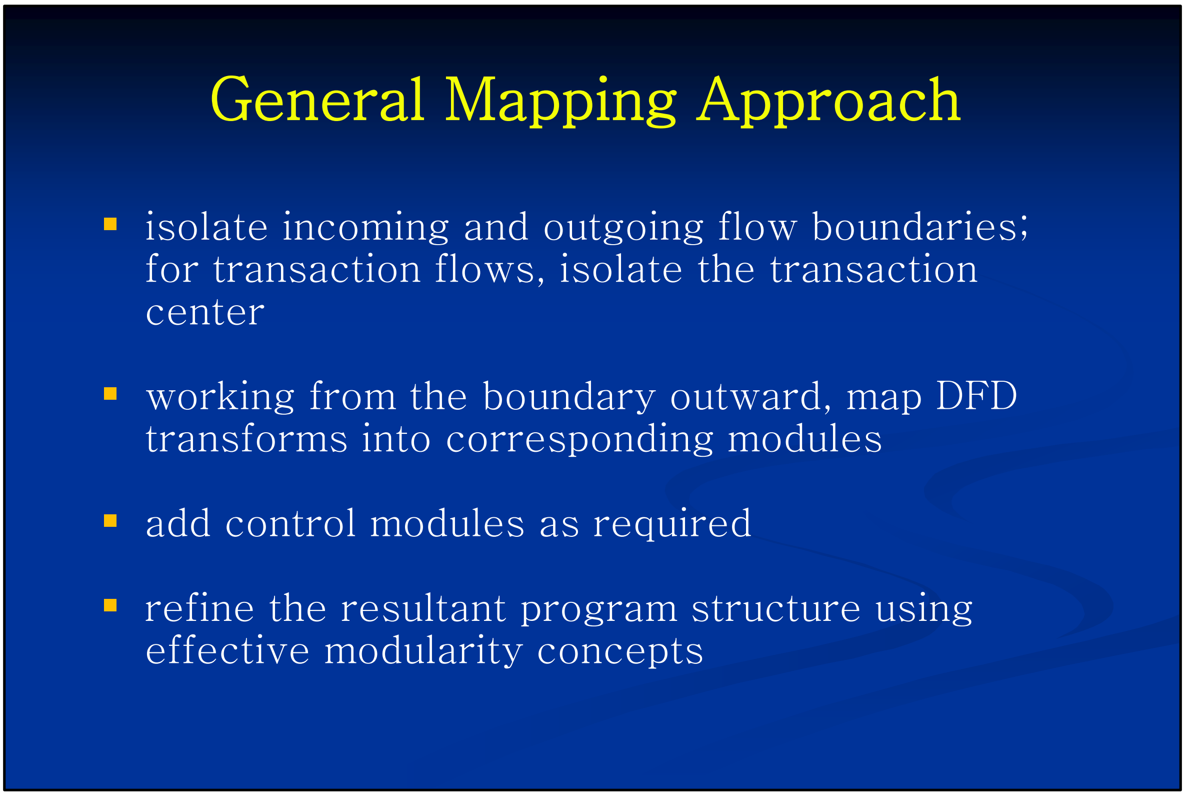
- results in software that is easier to test
- leads to software that is easier to maintain
- results in propagation of fewer side effects
- results in software that is easier to extend

Two Types of Flow Characteristics



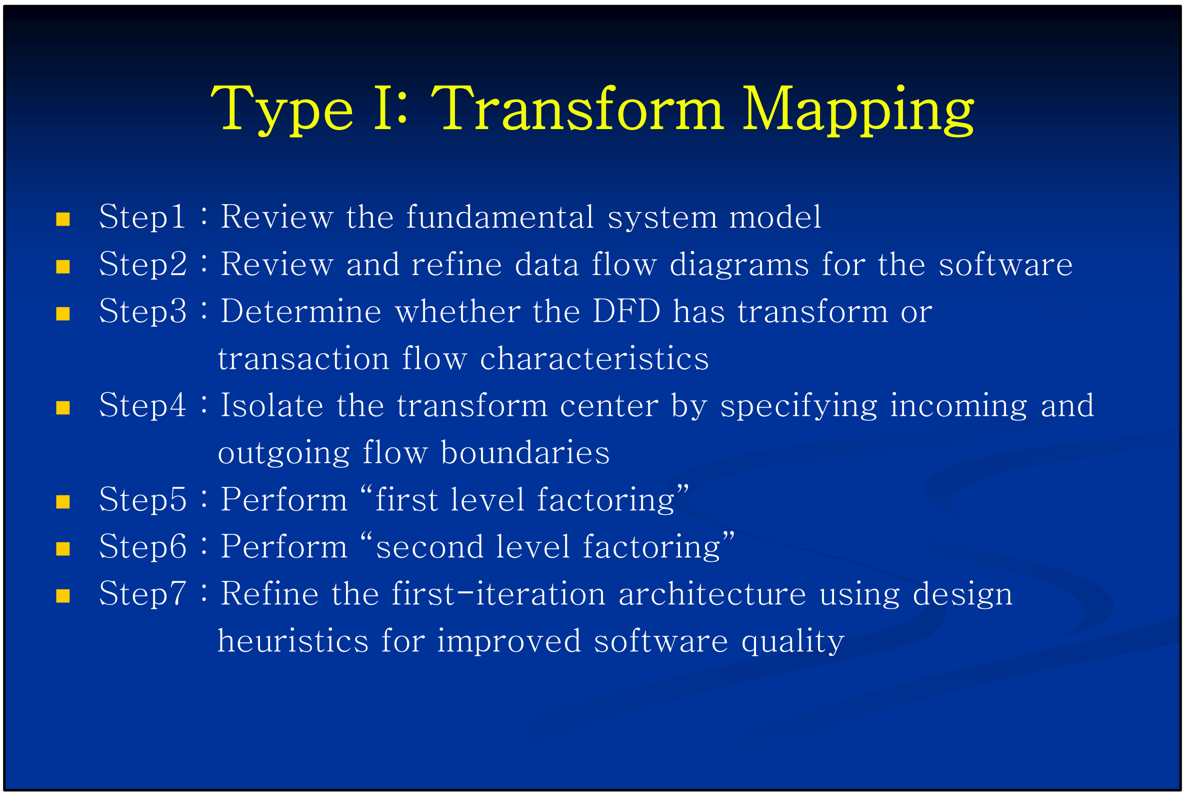
Type I:
Transform Flow

Type II:
Transaction Flow



General Mapping Approach

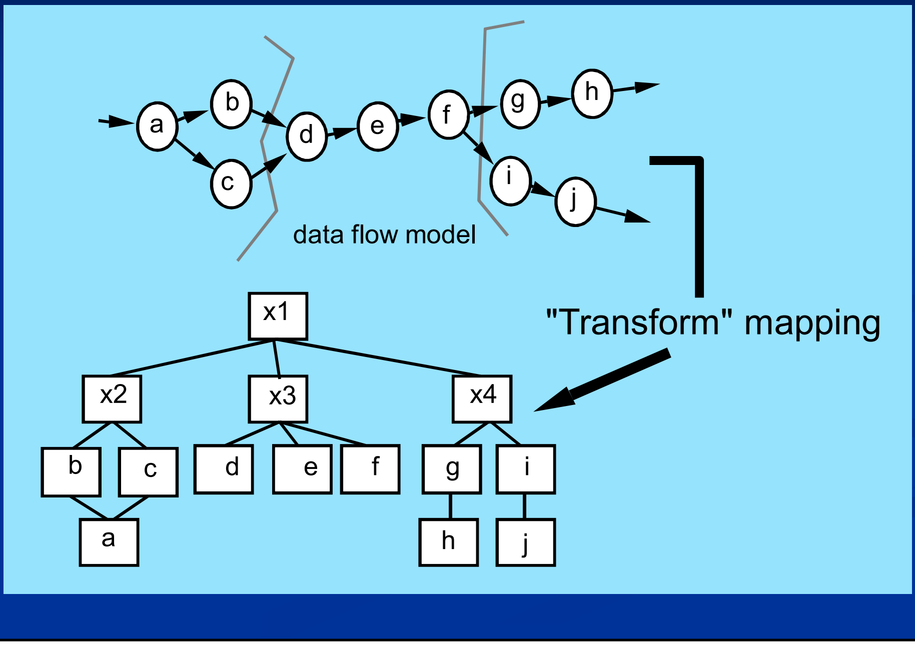
- isolate incoming and outgoing flow boundaries; for transaction flows, isolate the transaction center
- working from the boundary outward, map DFD transforms into corresponding modules
- add control modules as required
- refine the resultant program structure using effective modularity concepts



Type I: Transform Mapping

- Step1 : Review the fundamental system model
- Step2 : Review and refine data flow diagrams for the software
- Step3 : Determine whether the DFD has transform or transaction flow characteristics
- Step4 : Isolate the transform center by specifying incoming and outgoing flow boundaries
- Step5 : Perform "first level factoring"
- Step6 : Perform "second level factoring"
- Step7 : Refine the first-iteration architecture using design heuristics for improved software quality

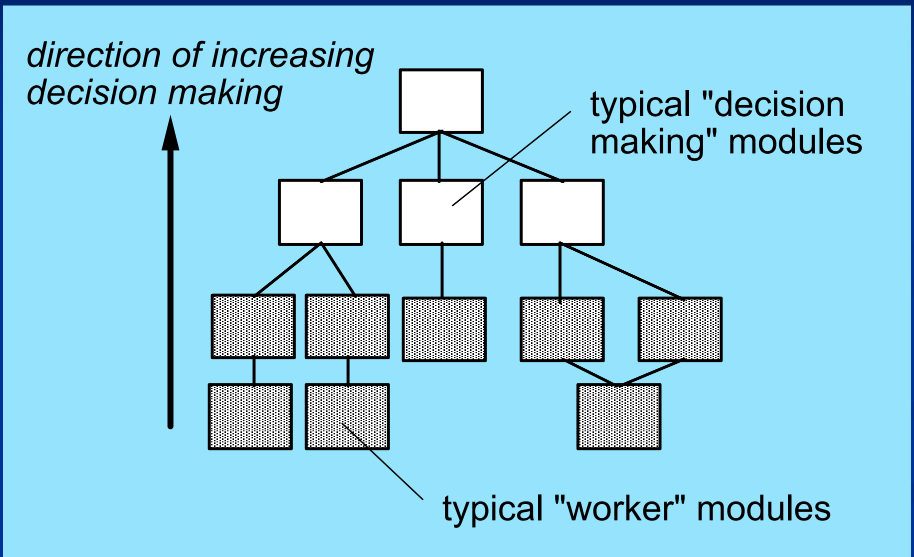
Transform Mapping



data flow model

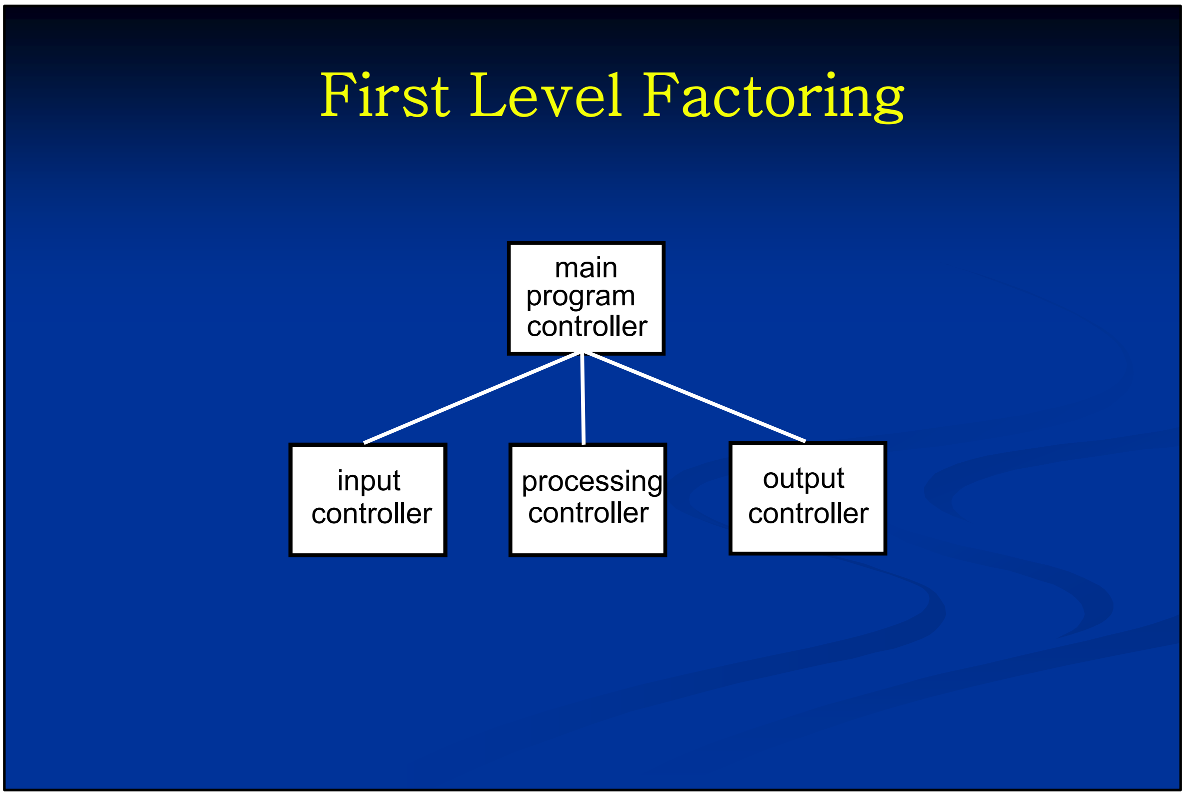
"Transform" mapping

Factoring


direction of increasing decision making

typical "decision making" modules

typical "worker" modules



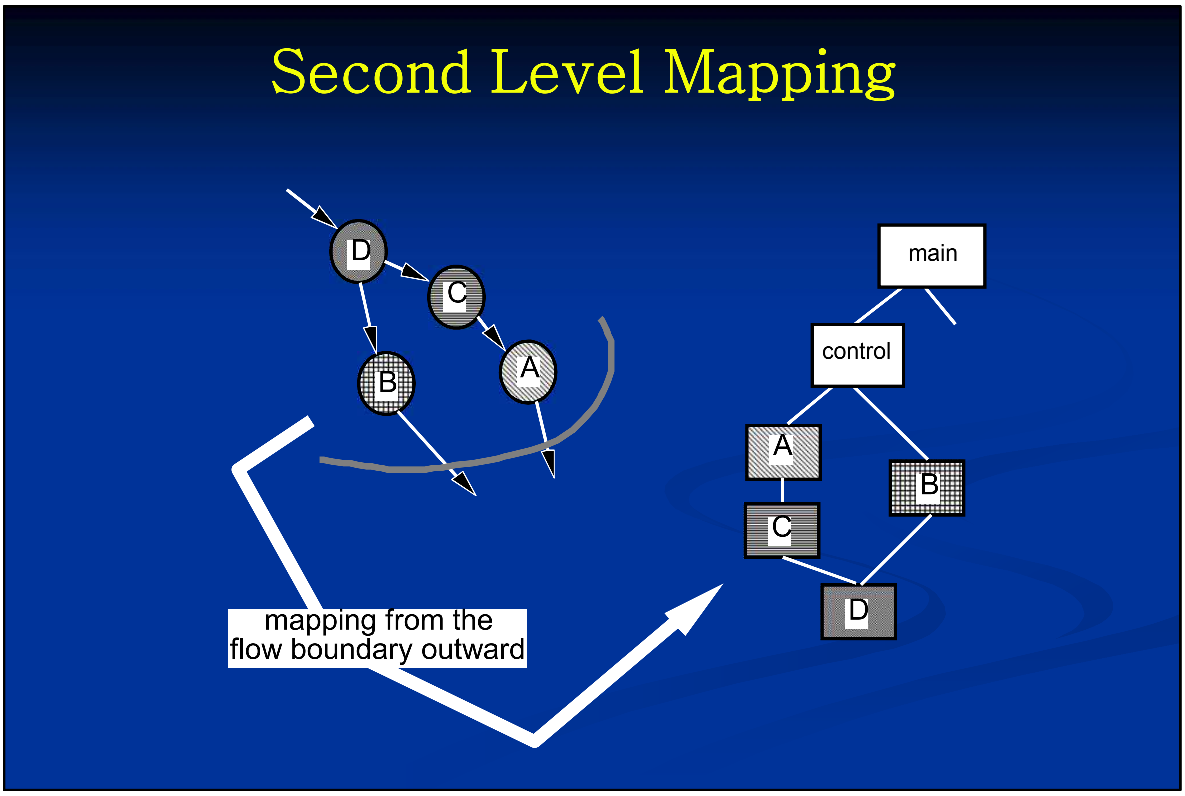
First Level Factoring

main program controller

input controller

processing controller

output controller



Second Level Mapping

D

C

B

A

main

control

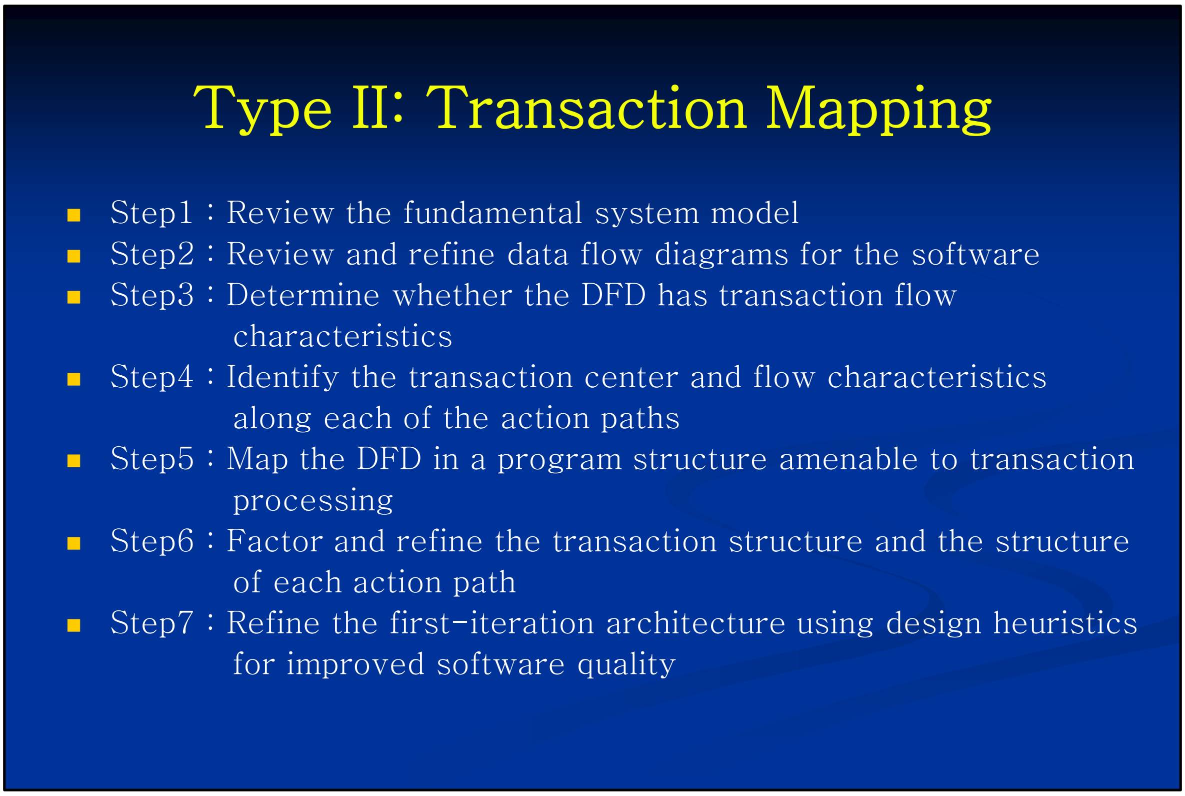
A

B

C

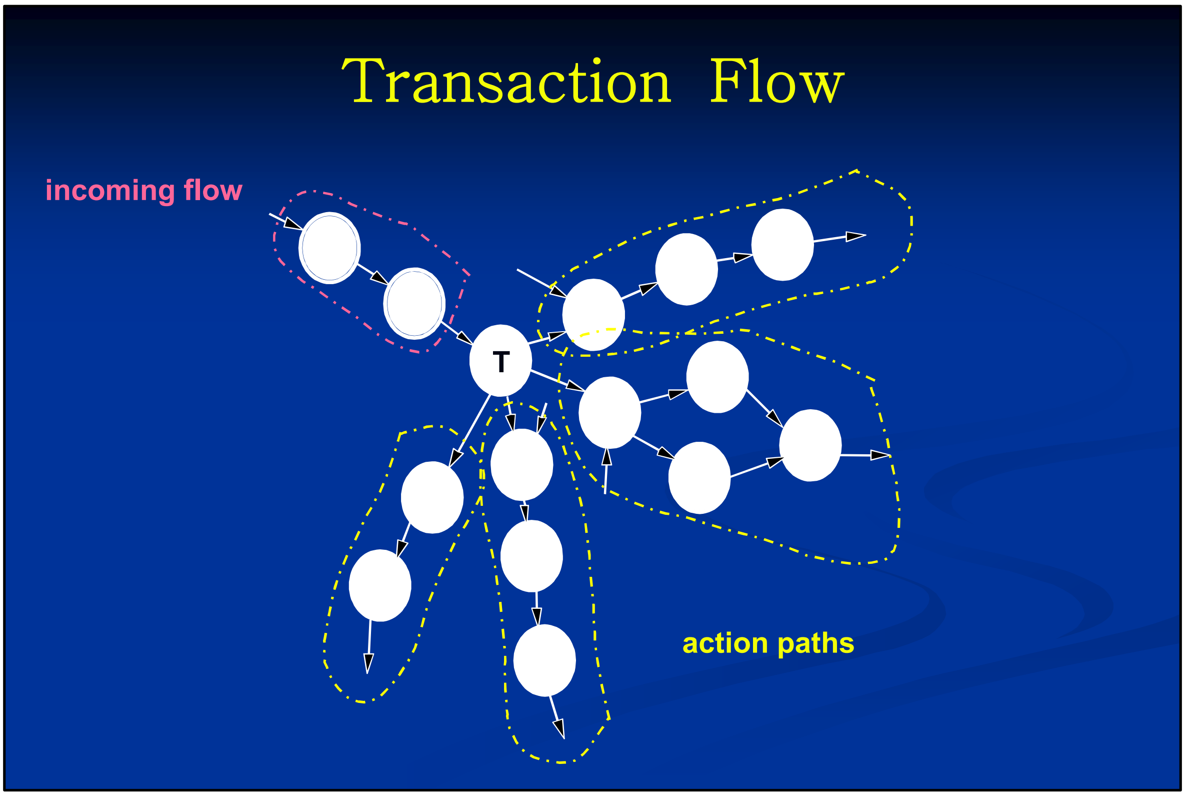
D

mapping from the flow boundary outward



Type II: Transaction Mapping

- Step1 : Review the fundamental system model
- Step2 : Review and refine data flow diagrams for the software
- Step3 : Determine whether the DFD has transaction flow characteristics
- Step4 : Identify the transaction center and flow characteristics along each of the action paths
- Step5 : Map the DFD in a program structure amenable to transaction processing
- Step6 : Factor and refine the transaction structure and the structure of each action path
- Step7 : Refine the first-iteration architecture using design heuristics for improved software quality



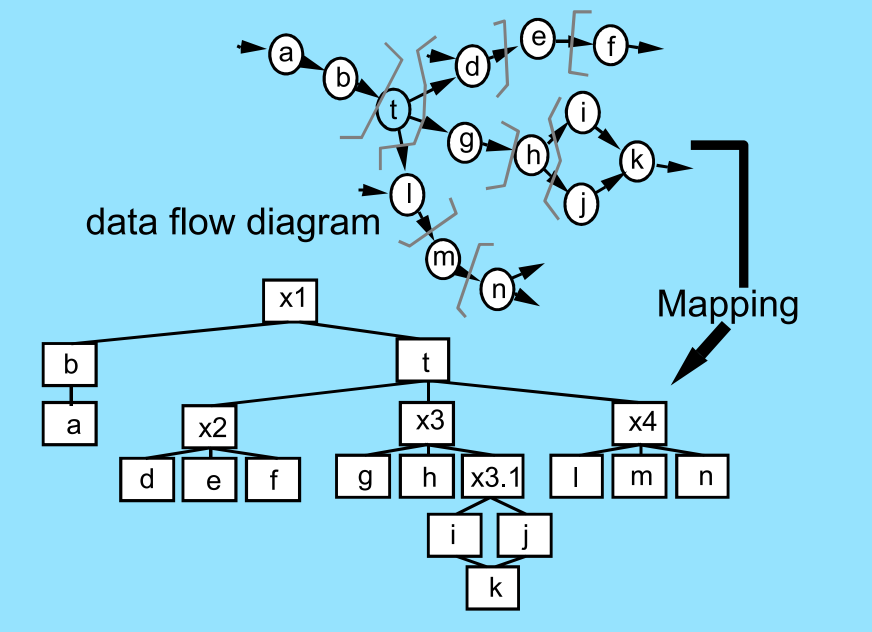
Transaction Flow

incoming flow

T

action paths

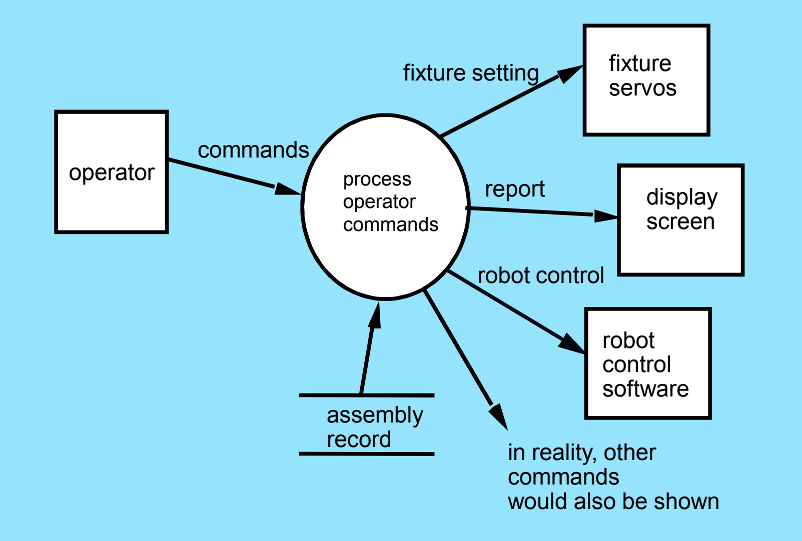
Transaction Mapping



data flow diagram

Mapping

Transaction Example



operator

commands

process operator commands

fixture setting

fixture servos

report

display screen

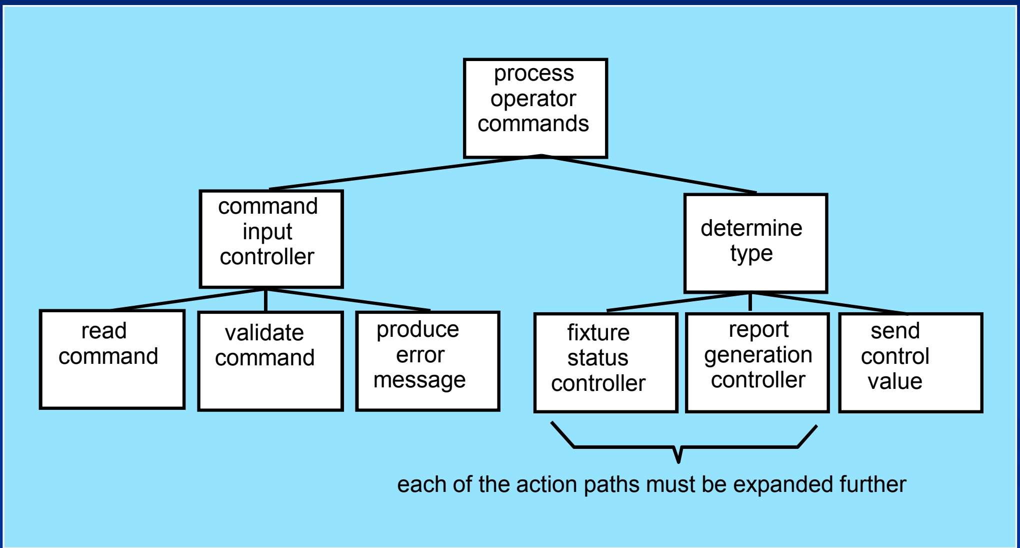
robot control

robot control software

assembly record

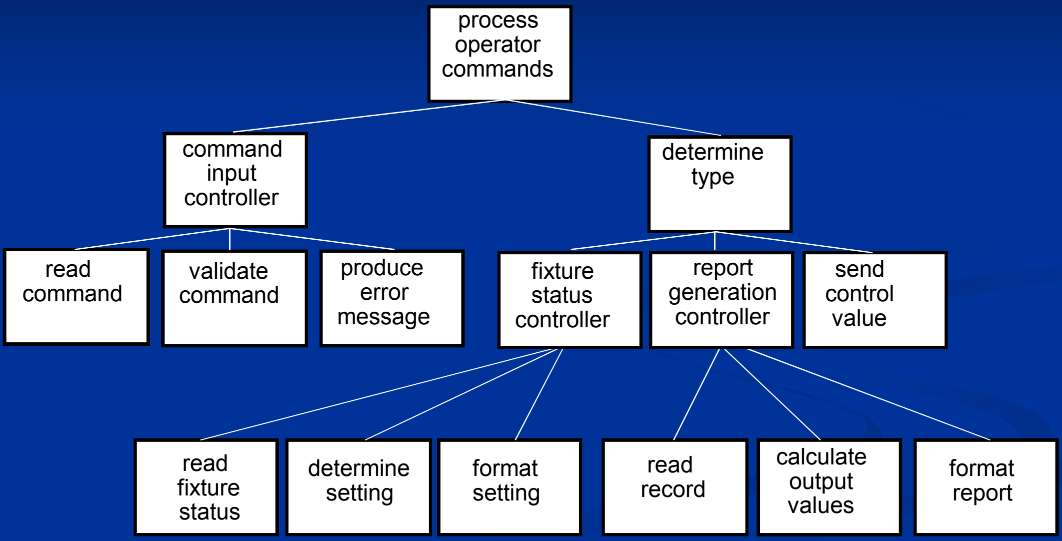
in reality, other commands would also be shown

Map the Flow Model



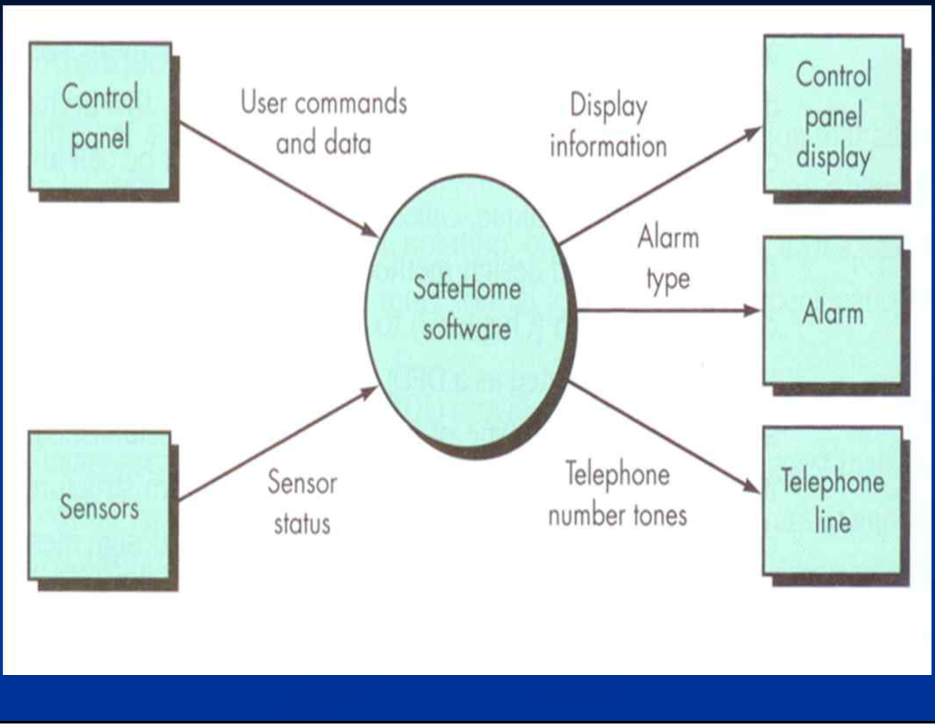
each of the action paths must be expanded further

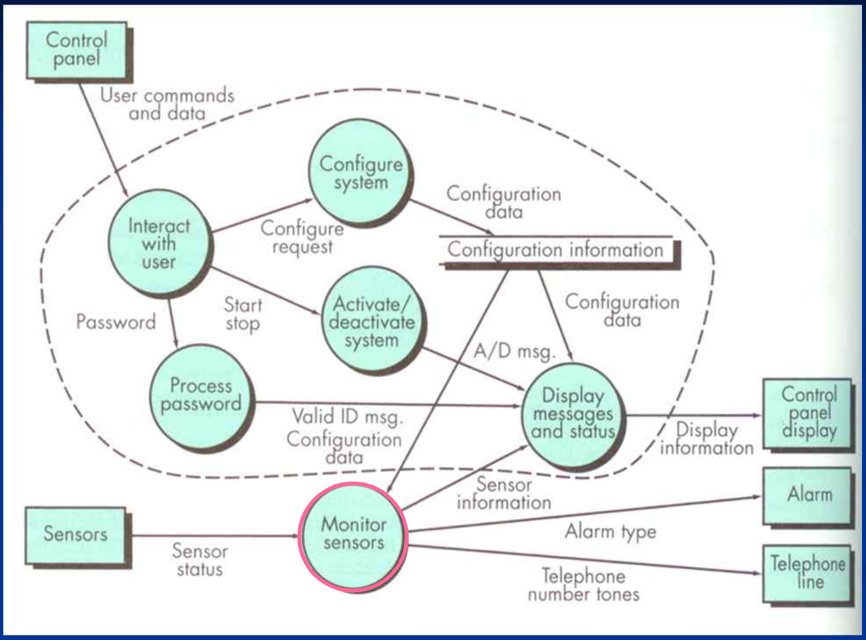
Refining the Structure Chart

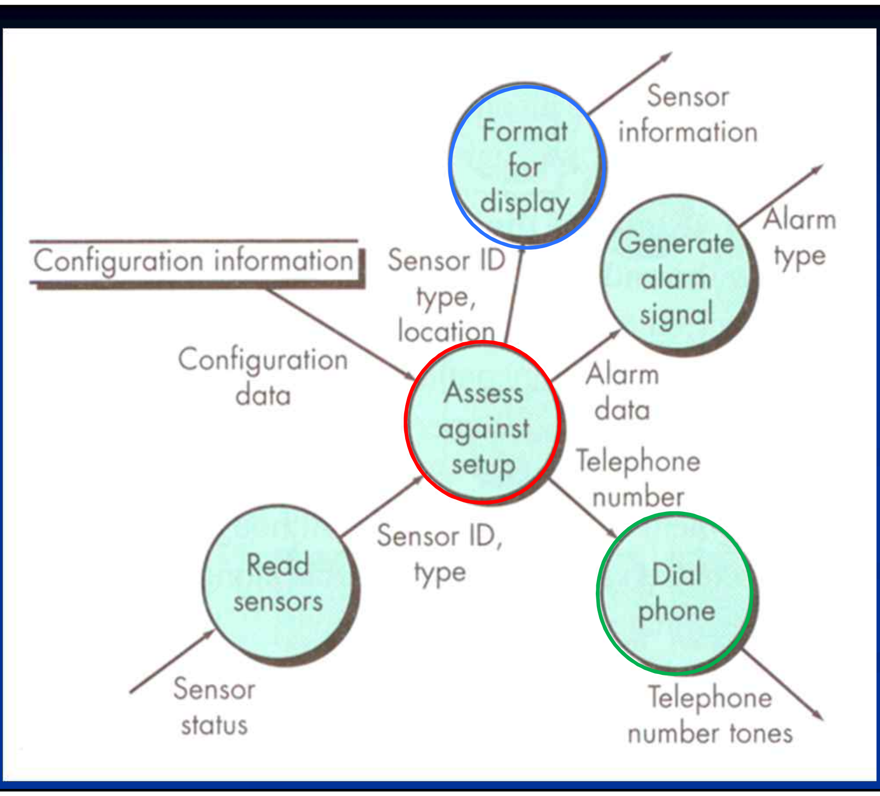


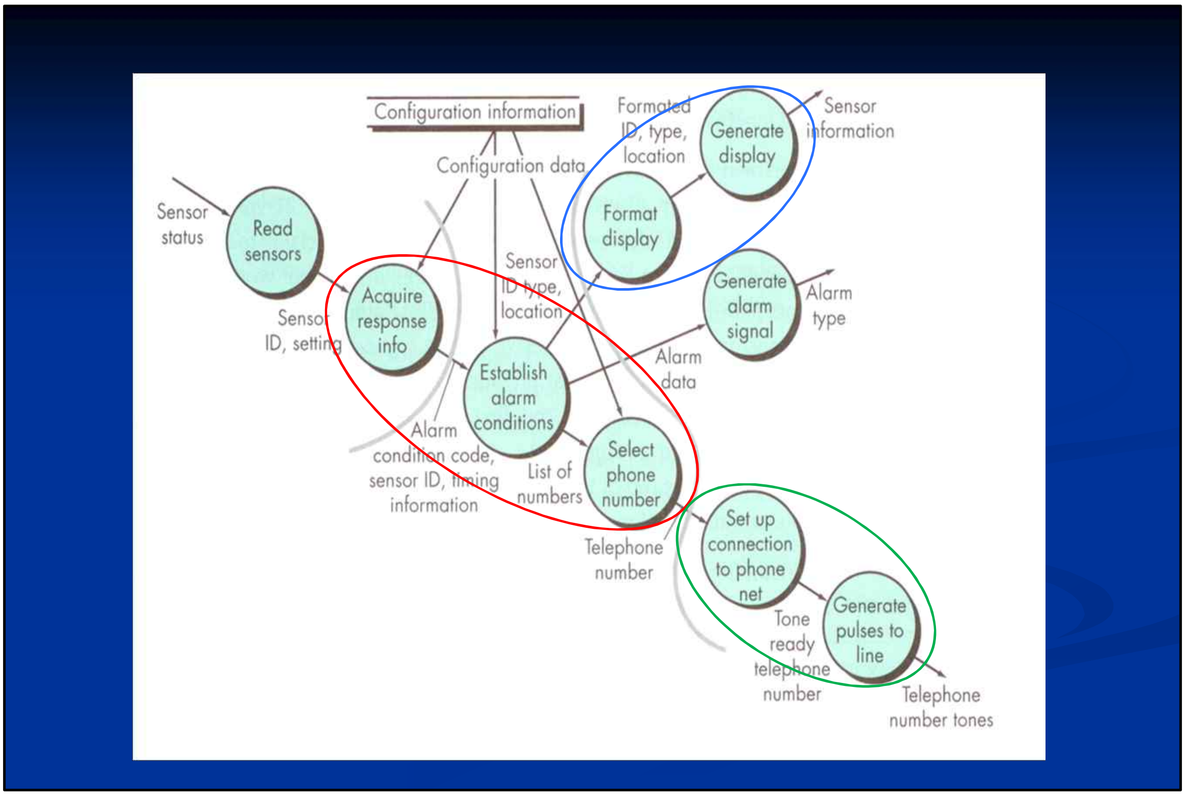
Refining the Architectural Design

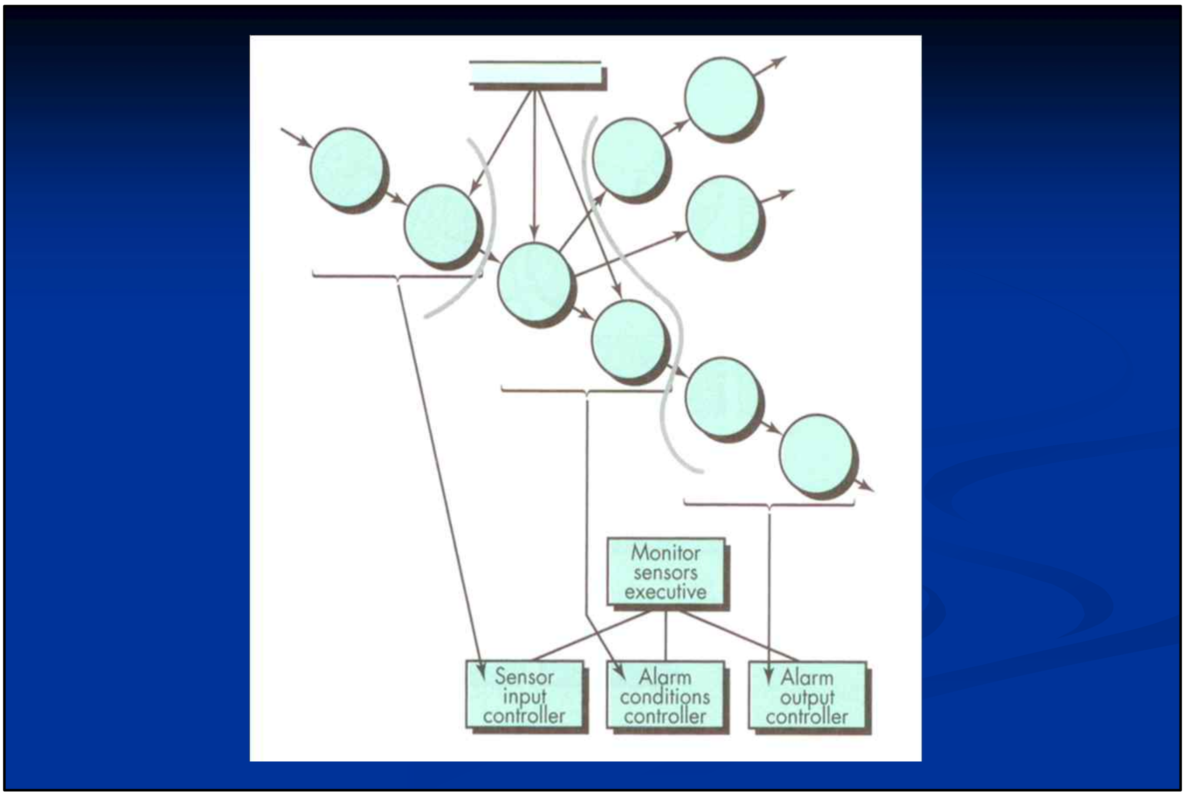
- Optimal design
- “best” approach (from alternatives)

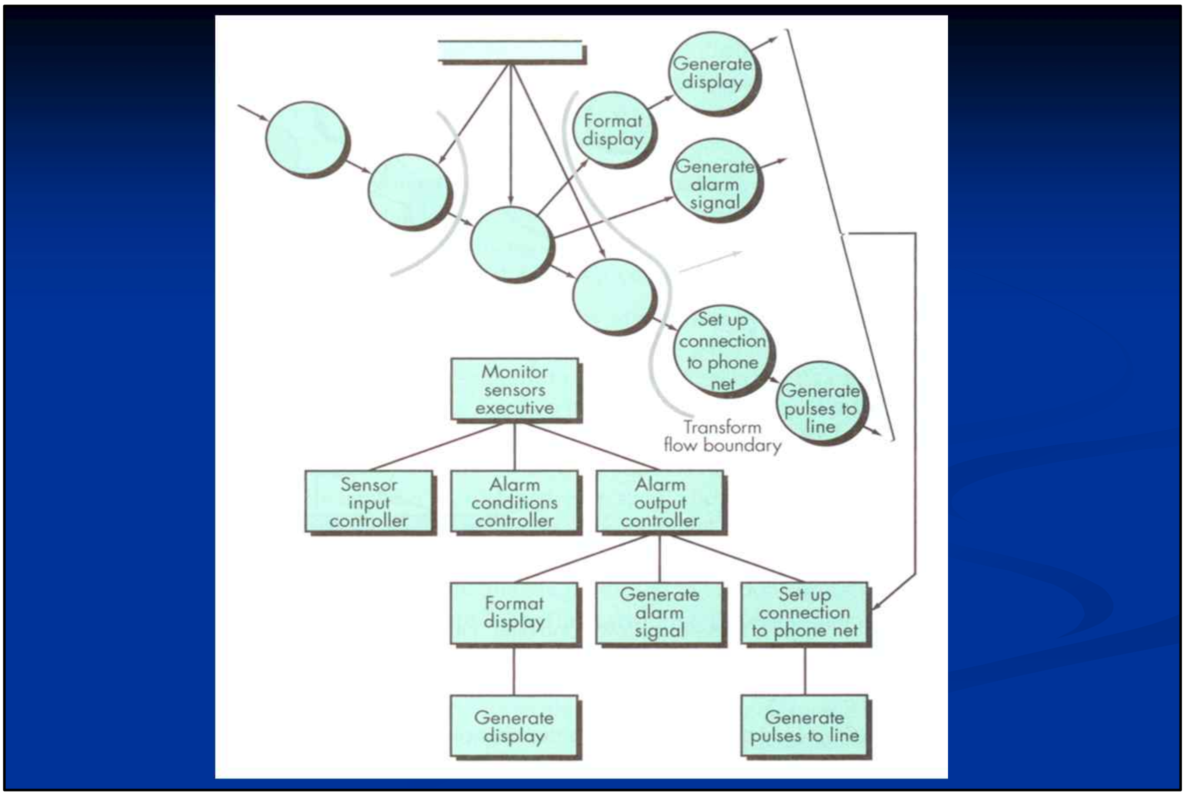


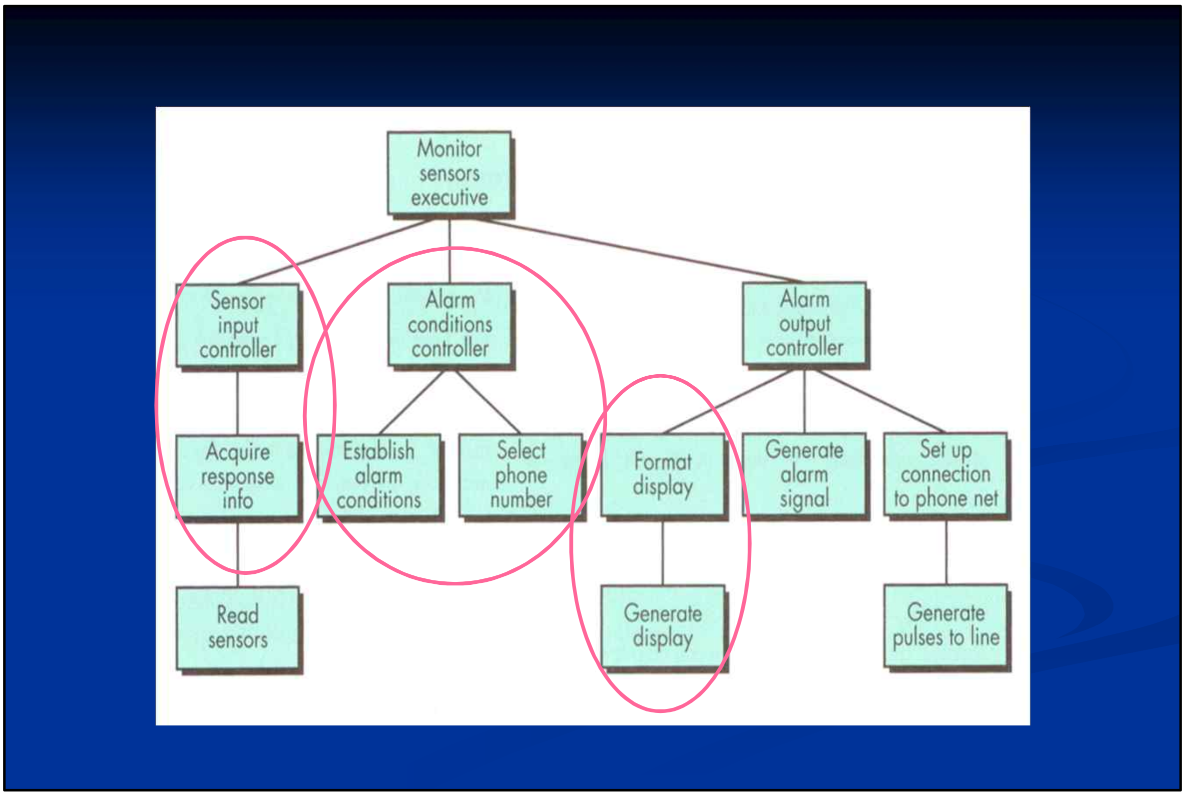

Control panel
User commands and data
Configure system
Configuration data
Interact with user
Configure request
Configuration information
Activate/ deactivate system
Configuration data
Password
Start stop
A/D msg.
Process password
Display messages and status
Control panel display
Valid ID msg.
Configuration data
Display information
Sensor information
Alarm
Sensors
Monitor sensors
Alarm type
Sensor status
Telephone number tones
Telephone line

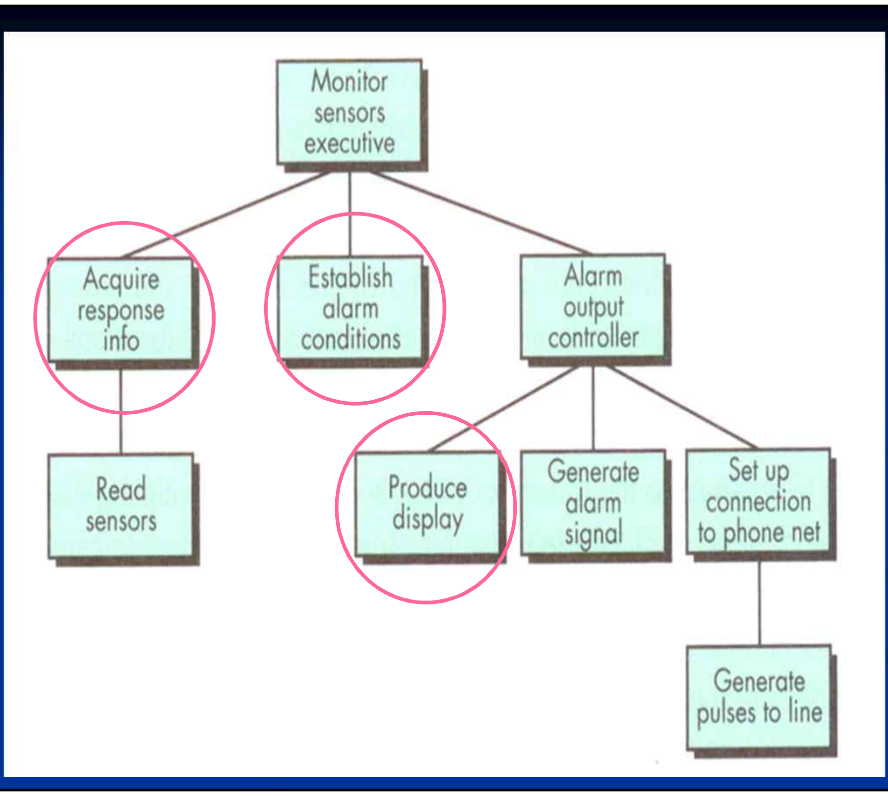

Sensor information
Format for display
Alarm type
Generate alarm signal
Configuration information
Sensor ID type, location
Configuration data
Assess against setup
Alarm data
Telephone number
Read sensors
Sensor ID, type
Dial phone
Sensor status
Telephone number tones


Configuration information
Configuration data
Sensor status
Read sensors
Sensor ID, setting
Acquire response info
Sensor ID type, location
Establish alarm conditions
Alarm condition code, sensor ID, timing information
List of numbers
Select phone number
Formated ID, type, location
Format display
Generate display
Sensor information
Generate alarm signal
Alarm type
Alarm data
Telephone number
Set up connection to phone net
Tone ready telephone number
Generate pulses to line
Telephone number tones


Monitor sensors executive
Sensor input controller
Alarm conditions controller
Alarm output controller


Generate display
Format display
Generate alarm signal
Set up connection to phone net
Generate pulses to line
Transform flow boundary
Monitor sensors executive
Sensor input controller
Alarm conditions controller
Alarm output controller
Format display
Generate alarm signal
Set up connection to phone net
Generate display
Generate pulses to line


Monitor sensors executive
Sensor input controller
Alarm conditions controller
Alarm output controller
Acquire response info
Establish alarm conditions
Select phone number
Format display
Generate alarm signal
Set up connection to phone net
Read sensors
Generate display
Generate pulses to line



Monitor sensors executive

Acquire response info

Establish alarm conditions

Alarm output controller

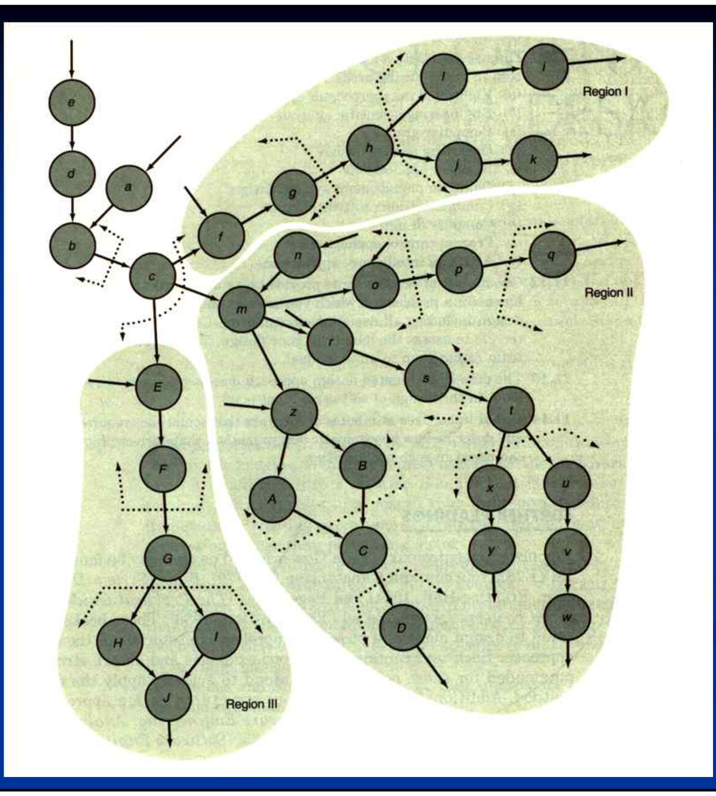
Read sensors

Produce display

Generate alarm signal

Set up connection to phone net

Generate pulses to line


Region I
Region II
Region III
e
d
a
b
c
f
g
h
i
l
j
k
n
m
o
p
q
r
s
t
x
u
y
v
w
z
B
A
C
D
E
F
G
H
I
J