Guidelines for transforming DFD models to Structure Chart models
The Reasons

- To move from an abstract system representation to a more physical one.
- To offer some guidelines to this procedure.
- To reduce ambiguity which may arise from subjective interpretations.
- To move from data flow concepts to program structure concepts.
General Steps Involved

1. The type of data flow is established
   - What is the nature of the data flowing between processes?

2. Determine flow boundaries (switch points)
   - Input↔Output boundaries
   - Hub↔Action boundaries

3. Map the abstract DFD on to a particular program structure
   - Transformational structure
   - Transactional structure
4. Define a valid control structure
   - Also known as “first-level” factoring
   - Depends on whether transformational or transactional models are used.
   - “Call-and-return” for transformational
   - “Call-and-act” for transactional

5. Refine (tune) the resulting structure
   - Also known as “second-level factoring”
   - Map IO flow bounded parts of DFD
6. Supplement and tune the final architectural structure

- Apply basic module independence concepts (i.e. Explode or implode modules according to coupling/cohesion requirements) to obtain an easier implementation.

- Please also visit the slides on the web-site “www.sei.cmu.edu/ata/ATAM/index.htm” for a more comprehensive and interesting approach to architectural analysis known as “Architectural Trade-off Analysis – ATA”.

Transformational Analysis (aka Transformational Mapping)

Context level (level 0) Example

- Control panel
- Sensors
- SafeHome system*
- Control panel display
- Alarm
- Telephone line

* Example taken from “Software Engineering – A Practitioner's Approach” by R. S. Pressman.
Level 2 Example (Monitor sensors)

1. **Assess against setup**
   - Configuration data
   - Sensor id, type and location
   - Sensor status and setting

2. **Read sensors**
   - Sensor id and setting

3. **Format for display**
   - Sensor information
   - Format for display
   - Alarm data

4. **Gen. alarm signal**
   - Alarm type
   - Telephone number

5. **Dial phone**
   - Dial tones
   - Dial phone
   - Sensor id, type and setting

6. **Telephone number**
   - Dial tones
   - Configuration data

7. **Diagonals**
   - Sensor status
   - Alarm data
   - Configuration data
Level 3 Example (Assess against setup)

- Acquire response info.
- Estab. alarm conds.
- Select phone number

- Sensor id and setting
- Alarm cond. Code, sensor id, and timing info.
- List of numbers
- Telephone number

- Sensor id, type and location
- Alarm data
- Configuration data
Level 3 Example (Format for display)

Sensor id, type and location  \rightarrow  Format display  \rightarrow  Formatted id, type and location  \rightarrow  Generate display  \rightarrow  Sensor information
Level 3 Example (Dial phone)

1. Telephone number
2. Setup conn to phone net
3. Tone-ready telephone number
4. Gen. pulses to line
5. Dial tones
Putting Level 3 Together

This DFD exhibits definite transform flow character.

- Afferent branch (input)
- Central transform (processing)
- Efferent branch (output)
First-Level Factoring

Monitor sensors

Sensor input controller

Alarm conditions controller

Alarm output controller
These are all the processes in the efferent branch:
Second-Level Factoring (2)

For the efferent branch:

- Monitor sensors
  - Sensor input controller
  - Alarm conditions controller
  - Alarm output controller
    - Format display
    - Generate alarm signal
    - Setup conn. to phone net
      - Generate pulses to line

Now, do the same for the other branches (i.e. Afferent and Central)...
Second-Level Factoring

- Estab. alarm conds.
- Configuration data
- List of numbers
- Sensor id, type and location
- Telephone number
- Alarm output controller
- Alarm cond. controller
- Sensor input controller
- Establish alarm conds
- Select phone number
- Select phone number
- Monitor sensors

For central transform:
Finally, for the afferent branch:

- Read sensors
  - Sensor status
  - Sensor id and setting
- Acquire response info.
  - Configuration data
  - Alarm cond. Code, sensor id, and timing info.
- Monitor sensors
  - Sensor input controller
  - Alarm conds controller
  - Alarm output controller
  - Acquire response info.
  - Read sensors
Some degree of refinement can sometimes be carried out on the initial “rough-cut” of the system's structure.

- Monitor sensors
  - Implode & assimilate (inherent functionality)
  - Sensor input controller
    - Read sensors
      - Implode (single flow)
  - Establish alarm conditions
  - Alarm output controller
    - Generate alarm signal
      - Generate pulses to line
    - Setup connection to phone net
      - Implode (triviality and single flow)
Transactional Analysis (aka Transactional Mapping)

- First step of this analysis is exactly the same as for transformational – i.e. refine the DFD.
- The initial “SafeHome*” example will be used.
- The level one DFD will be decomposed to level two as necessary.
- A “transactional centre” will then be determined.

* Taken from “Software Engineering – A Practitioner's Approach” by R. S. Pressman.
Level 2 Example (Interact with user)
Level 2 Example  (Process password)

- Read password
- Compare password with file
- Produce error msg

Configuration data

Password
Four digits
Valid id msg
Invalid password
Error msg
Level 2 Example (Configure system)

Sys parameters and data

Configure request → Read system data → Build config file

Raw conf data

Configuration data
Putting Level 2 Together

This DFD exhibits transaction flow character.

Transaction centre (dispatching)
Transaction Structure Mapping

user interaction

- Read user cmd
- Invoke cmd processing
  - Enable/disable sys.
  - Configuration controller
  - Password controller
Fully refining the DFD could yield the following structure:

- **user interaction**
  - Read user cmd
  - Invoke cmd processing
    - Enable/disable sys.
      - Read system data
      - Build config file
    - Configuration controller
      - Password controller
        - Read password
        - Compare password with file
        - Produce error msg
    - Display msgs & status