DATA FLOW DIAGRAMS

An Example

Remember during Composition

- a process should not have identical inputs and outputs
- all data flows must be labelled
- sources / sinks are not processes (=active)
- data stores should not be connected directly to sources / sinks
- there is no timing in a DFD

Suggestions for a Good Style

- organise diagram from left to right or top to bottom
- provide input and output “filters” for each major data flow
- keep sources / sinks on the left or right boundary of your diagram

Process Decomposition

- processes can be decomposed / refined
- one process ==> complete DFD
- interface flows must remain consistent
- lower level processes, data flows and data stores can be added on
- sources sinks remain on level-1
- a level-0 can be used as “abstract”

Process Decomposition

Interface Consistency

End of Section 4b

coming up:
data dictionaries
The DATA DICTIONARY
(for DFDs)

Concepts and Examples

Data Dictionary

- part of the specifications for a complete
  - Dataflow Analysis (DFD)
  - ER model
  - OOAD model
  - SADT
  - etc.

Data Dictionary for a DFD

- specifies data elements in a DFD:
  - data flows
  - simple data stores
- must be consistent with DFD
  - specification compatible with diagram
- must be complete with regard to the DFD
- consists of data item specifications

Data Item Specification

- name (and aliases)
- informal description
- purpose
- range of values
- related data items
- connections to processes (flow information)
- formal data structure specification

Patient Data Record

- name: patient data record (PDR)
- description:
  A PDR is produced by the Patient Monitoring System and includes information on the current status of a specific patient in the intensive care unit, e.g. blood pressure, temperature, etc.
  A PDR is produced every five minutes.

Patient Data Record

- purpose:
  Used to feed the Patient DB with up-to-date information on each patient.
PDRs are accumulated, i.e. the average value of all received readings during a full hour is stored.
- range of values:
  see specifications of PDR sub-elements
Patient Data Record

- related data items:
  - derived items:
    - PDR
    - Patient Input
  - is-part-of:
    N/A
    (PDR has no super-ordinate element)

- is-decomposed-into (continued):
  - status (critical or non-critical)
  - delta values for
    heart-rate, temperature, blood-pressure
    as calculated during the last hour
    comparing to the full hour value
    (minimum 1, maximum 12)

Patient Data Record

- related data items:
  - is-decomposed-into (sub-ordinate elem.):
    - patient-id
    - patient-name (first and last)
    - heart-rate
    - temperature
    - blood-pressure (optional)

- connections to processes:
  - comes from:
    Patient Monitoring System
    (external source)
  - goes to:
    check & accum. PDR
    (system internal process)

Formal Data Structure Definition?

- BNF - Backus-Naur-Form:
  - formal language
  - context free grammar
  - based on substitution rules
  - widely used to specify syntax of
    programming languages

BNF

- rules used to specify
  substitution/refinement/structuring:

  data-item  ==>  data-structure
  person  ==>  name + address
  name  ==>  first_name + last_name
BNF

- meta-symbols used to describe structure:

  + sequence

  [...][...][... ] exclusive alternatives

  { ... } iteration (>=1 repetitions)

  ( ... ) option

---

Patient Data Record - BNF

- formal structure specification:

  PDR ==> p_id + p_name + c_val + status + {d_val}
  p_name ==> first_name + last_name
  c_val ==> heart_rate + temp + ( blood_pressure )
  status ==> [ "critical" | "non-critical" ]
  d_val ==> d_heart + d_temp + ( d_blood )

---

End of Section 4c

coming up:
process specifications