

Analysis Modeling

Oleh :
Ir. I Gede Made Karma, MT

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

1

Analysis Modeling: Where to Begin?

- A statement of scope can be acquired from:
 - ⇨ the FAST working document
 - ⇨ A set of use-cases
- the statement of scope must be "parsed" to extract data, function and behavioral domain information

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

2

Statement of Scope

- a relatively brief description of the system to be built
 - ⇨ indicates data that are input and output and basic functionality
 - ⇨ indicates conditional processing (at a high level)
 - ⇨ implies certain constraints and limitations

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

3

Identifying Objects and Operations

- define "objects" by underlining all nouns in the written statement of scope
 - producers/consumers of data
 - places where data are stored
 - "composite" data items
- define "operations" by double underlining all active verbs
 - processes relevant to the application
 - data transformations
- consider other "services" that will be required by the objects

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

4

Data Modeling and Entity Relationship (E-R) Diagramming

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

5

Why Data Modeling?

- examines data objects independently of processing
- focuses attention on the data domain
- creates a model at the customer's level of abstraction
- indicates how data objects relate to one another

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

6

What is a Data Object?

Object—something that is described by a set of attributes (data items) and that will be manipulated within the software (system)

- each **instance** of an object (e.g., a book) can be identified uniquely (e.g., ISBN #)
- each plays a necessary role in the system i.e., the system could not function without access to instances of the object
- each is described by attributes that are themselves data items

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

7

Typical Objects

- external entities** (printer, user, sensor)
- things** (e.g., reports, displays, signals)
- occurrences or events** (e.g., interrupt, alarm)
- roles** (e.g., manager, engineer, salesperson)
- organizational units** (e.g., division, team)
- places** (e.g., manufacturing floor)
- structures** (e.g., employee record)

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

8

Data Objects and Attributes

A data object contains a set of attributes that act as an aspect, quality, characteristic, or descriptor of the object

<p>object: automobile</p> <p>attributes: make model body type price options code</p>
--

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

9

What is a Relationship?

relationship—indicates “connectedness”; a “fact” that must be “remembered” by the system and cannot or is not computed or derived mechanically

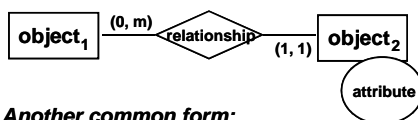
- several instances of a relationship can exist
- objects can be related in many different ways

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

10

ERD Notation

One common form:



Another common form:



These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

11

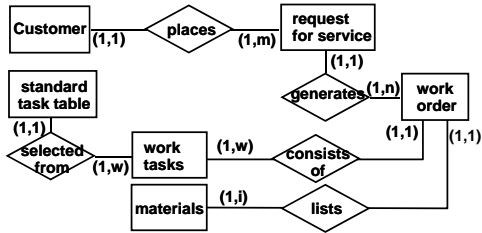
Building an ERD

- Level 1**—model all data objects (entities) and their “connections” to one another
- Level 2**—model all entities and relationships
- Level 3**—model all entities, relationships, and the attributes that provide further depth

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

12

The ERD: An Example



These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach, 5/e* and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

13

Creating a Flow Model

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach, 5/e* and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

14

The Flow Model

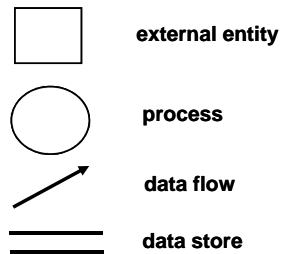
Every computer-based system is an information transform



These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach, 5/e* and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

15

Flow Modeling Notation



These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach, 5/e* and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

16

External Entity

A producer or consumer of data

Examples: a person, a device, a sensor

Another example: computer-based system

Data must always originate somewhere and must always be sent to something

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach, 5/e* and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

17

Process

A data transformer (changes input to output)

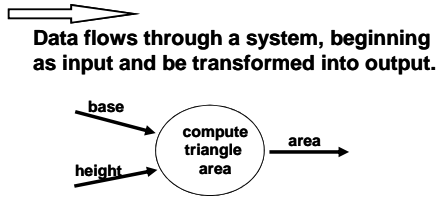
Examples: compute taxes, determine area, format report, display graph

Data must always be processed in some way to achieve system function

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach, 5/e* and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

18

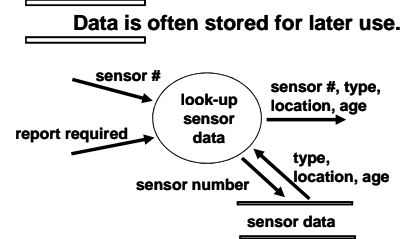
Data Flow



These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

19

Data Stores



These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

20

Data Flow Diagramming: Guidelines

- all icons must be labeled with meaningful names
- the DFD evolves through a number of levels of detail
- always begin with a context level diagram (also called level 0)
- always show external entities at level 0
- always label data flow arrows
- do not represent procedural logic

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

21

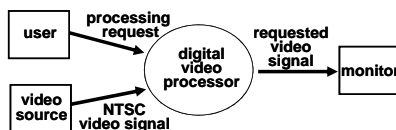
Constructing a DFD—I

- review ERD to isolate data objects and grammatical parse to determine operations)
- determine external entities (producers and consumers of data)
- create a level 0 DFD

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

22

Level 0 DFD Example



These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

23

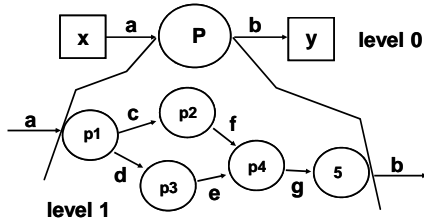
Constructing a DFD—II

- write a narrative describing the transform
- parse to determine next level transforms
- "balance" the flow to maintain data flow continuity
- develop a level 1 DFD
- use a 1:5 (approx.) expansion ratio

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

24

The Data Flow Hierarchy



These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

25

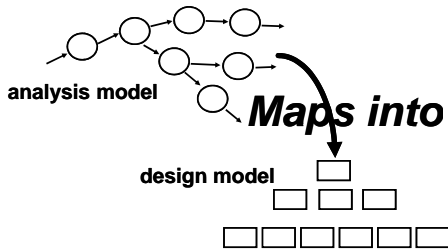
Flow Modeling Notes

- each bubble is refined until it does just one thing
- the expansion ratio decreases as the number of levels increase
- most systems require between 3 and 7 levels for an adequate flow model
- a single data flow item (arrow) may be expanded as levels increase (data dictionary provides information)

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

26

DFDs: A Look Ahead



These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

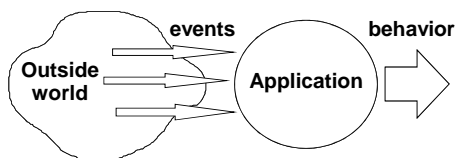
27

Behavioral Modeling and Process Specification

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

28

Behavioral Modeling



These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

29

The States of a System

- state—a set of observable circumstances that characterizes the behavior of a system at a given time
- state transition—the movement from one state to another
- event—an occurrence that causes the system to exhibit some predictable form of behavior
- action—process that occurs as a consequence of making a transition

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

30

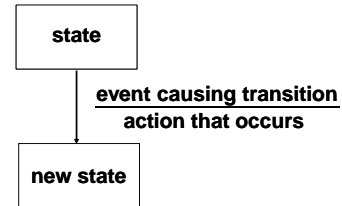
Behavioral Modeling

- make a list of the different states of a system (How does the system behave?)
- indicate how the system makes a transition from one state to another (How does the system change state?)
 - \Leftarrow indicate event
 - \Rightarrow indicate action
- draw a state transition diagram

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach, 5/e* and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

31

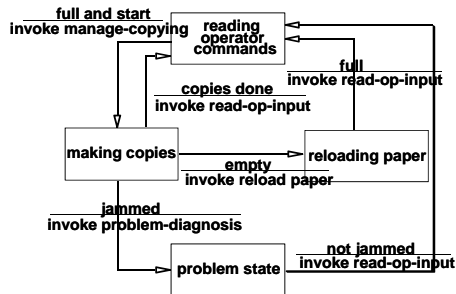
State Transition Diagram Notation



These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach, 5/e* and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

32

State Transition Diagram



These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach, 5/e* and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

33

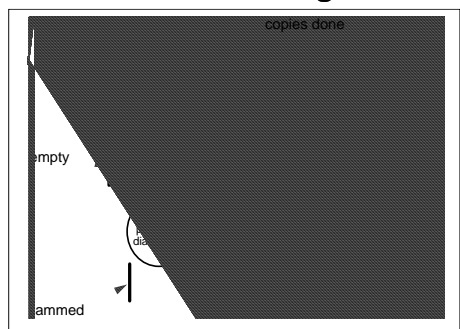
The Control Model

- the control flow diagram is "superimposed" on the DFD and shows events that control the processes noted in the DFD
- control flows—events and control items—are noted by dashed arrows
- a vertical bar implies an input to or output from a control spec (CSPEC) — a separate specification that describes how control is handled
- a dashed arrow entering a vertical bar is an input to the CSPEC
- a dashed arrow leaving a process implies a data condition
- a dashed arrow entering a process implies a control input read directly by the process
- control flows do not physically activate/deactivate the processes—this is done via the CSPEC

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach, 5/e* and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

34

Control Flow Diagram



These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach, 5/e* and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

35

Control Specification (CSPEC)

The CSPEC can be:

- state transition diagram (sequential spec)
 - state transition table
 - decision tables
 - activation tables
- } combinatorial spec

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach, 5/e* and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

36

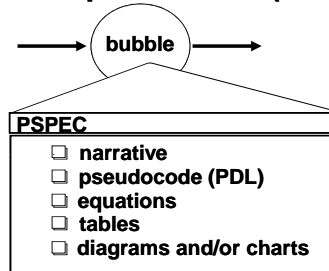
Guidelines for Building a CSPEC

- list all sensors that are "read" by the software
- list all interrupt conditions
- list all "switches" that are actuated by the operator
- list all data conditions
- recalling the noun-verb parse that was applied to the software statement of scope, review all "control items" as possible CSPEC inputs/outputs
- describe the behavior of a system by identifying its states; identify how each state is reach and defines the transitions between states
- focus on possible omissions ... a very common error in specifying control, e.g., ask: "Is there any other way I can get to this state or exit from it?"

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

37

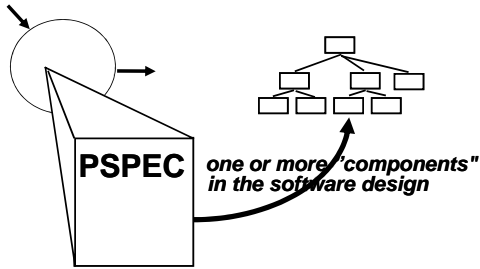
Process Specification (PSPEC)



These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

38

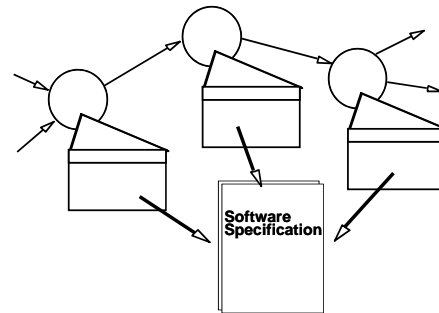
A Design Note



These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

39

Creating Mini-Specs



These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

40

Real-Time Analysis Methods

- each introduces its own notation, but all
 - ↔ propose an approach for representing control and separating it from data
 - ↔ provide a mechanism for specifying events, states, and distributed processing
 - ↔ begin at the analysis level and work to derive processor assignments, tasks and program architectures

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

41

Real-Time Analysis & Design Methods

- Gomaa, H., *Software Design Methods for Concurrent and Real-Time Systems*, Addison-Wesley, 1995.
- Harel, D. et al., "STATEMATE: A Working Environment for the Development of Complex Reactive Systems, *IEEE Trans. Software Engineering*, vol. 16, no. 3, April, 1990, pp. 403-414.
- Hatley, D.J. and I.A. Pirbhai, *Strategies for Real-Time System Specification*, Dorset House, 1987.
- Selic, B., G. Gullekson, and P. Ward, *Real-Time Object-Oriented Modeling*, Wiley, 1994.
- Shumate, K. and M. Keller, *Software Specification and Design—A Disciplined Approach For Real-Time Systems*, Wiley 1992.
- Ward, P.T. and S.J. Mellor, *Structured Development for Real-Time Systems*, 3 volumes, Yourdon Press, 1985, 1986.

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

42

The Data Dictionary

- ❑ a quasi-formal grammar for describing the content of data that the software will process and create
- ❑ a notation for describing control data and the values that control data can take, e.g., "on," or "off"
- ❑ a repository that also contains "where-used" / "how used" information
- ❑ a notation that can be represented manually, but is best developed using CASE tools

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

43

Building a Data Dictionary

Name:	the primary name of the composite data item
Aliases:	other names for the data item
Where used:	data transforms (processes) that use the composite data item
How used:	the role of the data item (input, output, temporary storage, etc.)
Description:	a notation for representing content (presented on next slide)
Format:	specific information about data types, pre-set values (if known)

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

44

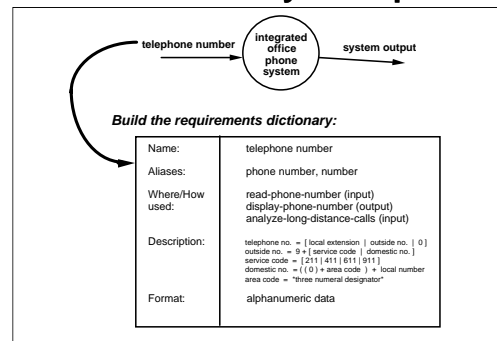
Data Dictionary Notation

<i>Notation</i>	<i>Meaning</i>
=	is composed of
+	and
[]	either-or
{ } ⁿ	n repetitions of
(...)	optional data
* ... text ...*	delimits a comment

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

45

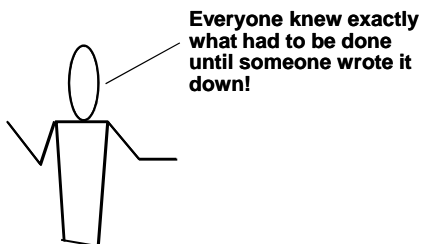
Data Dictionary Example



These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

46

Writing the Software Specification



These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

47

Specification Guidelines

- ❑ use a layered format that provides increasing detail as the "layers" deepen
- ❑ use consistent graphical notation and apply textual terms consistently (stay away from aliases)
- ❑ be sure to define all acronyms
- ❑ be sure to include a table of contents; ideally, include an index and/or a glossary
- ❑ write in a simple, unambiguous style (see "editing suggestions" on the following pages)
- ❑ always put yourself in the reader's position, "Would I be able to understand this if I wasn't intimately familiar with the system?"

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

48

Specification Guidelines

Be on the lookout for persuasive connectors, ask why?

keys: *certainly, therefore, clearly, obviously, it follows that ...*

Watch out for vague terms

keys: *some, sometimes, often, usually, ordinarily, most, mostly ...*

When lists are given, but not completed, be sure all items are understood

keys: *etc., and so forth, and so on, such as*

Be sure stated ranges don't contain unstated assumptions

e.g., *Valid codes range from 10 to 100. Integer? Real? Hex?*

Beware of vague verbs such as *handled, rejected, processed, ...*

Beware "passive voice" statements

e.g., *The parameters are initialized. By what?*

Beware "dangling" pronouns

e.g., *The I/O module communicated with the data validation module and its control flag is set. Whose control flag?*

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

49

Specification Guidelines

When a term is explicitly defined in one place, try substituting the definition for other occurrences of the term

When a structure is described in words, draw a picture

When a structure is described with a picture, try to redraw the picture to emphasize different elements of the structure

When symbolic equations are used, try expressing their meaning in words

When a calculation is specified, work at least two examples

Look for statements that imply certainty, then ask for proof keys; always, every, all, none, never

Search behind certainty statements—be sure restrictions or limitations are realistic

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner's Approach*, 5/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001

50