

# **CSC444F: Software Engineering I**

**Mou Hu**  
**[mou.hu@utoronto.ca](mailto:mou.hu@utoronto.ca)**

# Lecture 9: Software Design (II)

- Design Method 2 - Data Flow Design (SA/SD)
- Design Method 3 - Design Based on Data Structures (JSP & JSD)
- Reading: Chapter 11

# Design Method 2 - Data Flow Design (1/2)

- Data flow design is a functional decomposition with respect to the flow of data
- Data flow design process: two steps (SA and SD)
- A DFD is composed of external entities, processes, data flows, and data stores
- A structured chart describes modules and the call relations among them
- **Structure Analysis (SA)**
  - (1) At the highest level, draw a context diagram
  - (2) Decompose the context diagram to a data flow diagram (DFD)
  - (3) Further decompose the DFD to a lower level DFD
  - (4) Repeat (3) until every process becomes sufficiently straightforward and does not warrant further expansion

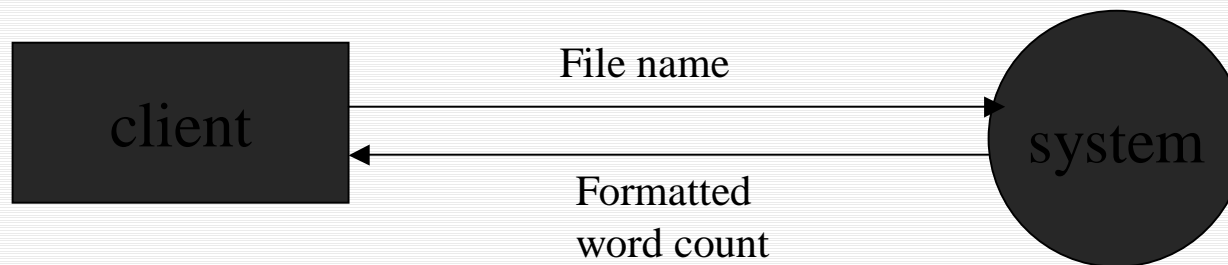
# Design Method 2 - Data Flow Design (2/2)

## ■ Structured Design (SD)

- (1) In the lowest level DFD, find the point of highest abstraction of input and the point of highest abstraction of output
- (2) Use the point of highest abstraction of input and the point of highest abstraction of output to transfer the DFD to a structure chart with a root module and three modules: the input module, transfer module, and output module
- (3) Decompose each module until each module performs a single action (with high cohesion)
- (4) Made necessary modification to achieve the lowest possible coupling among modules

# SA/SD Example

- Design a system that takes as input a file name and returns the number of words in that file, similar to the UNIX `wc` utility
- Context diagram



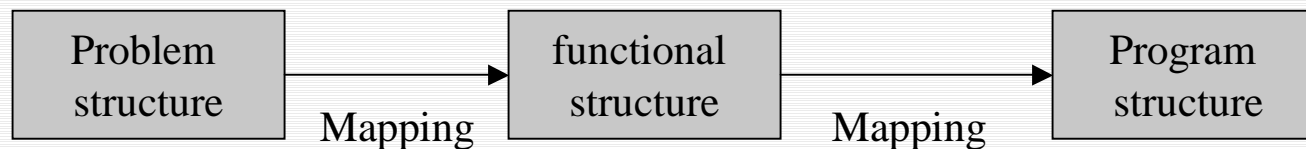
- See handout for DFDs and the design process.

# Design Method 3 - Design Based on Data Structures (JSP & JSD)

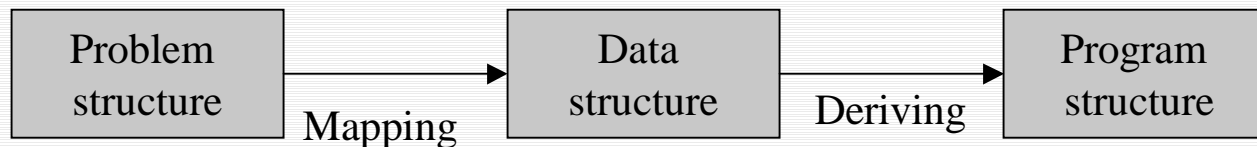
- Jackson Structured Programming (JSP) using structure diagrams to model data and program. The structure diagram can represent sequence, iteration, and selection.
- JSP design steps:
  1. Modeling input and output using structure diagrams
  2. Merging the diagrams to create the program structure
  3. Resolving possible structure clashes
  4. Optimizing the result through program inversion
- Jackson System Development (JSD):
  - Stage 1. Modeling (identifying entities and events, using a process structure diagram (PSD) to represent an entity)
  - Stage 2. Network (modeling system as a network of interconnected processes, using a system specification diagram (SSD) to represent the system)
  - Stage 3. Implementation (transforming SSD into an executable system)

# Design Method Comparison

■ In functional decomposition and data flow design:



■ In JSP and JSD:



Postulate: The structure of data is much less volatile than the transformations applied to the data. Design starting from data should be better.

■ JSP and JSD were never as popular as functional decomposition and data flow design and, with the rise of the object-oriented paradigm, largely has fallen out of fashion.

# References

- [1] Hans van Vliet, “Software Engineering: Principles and Practice”, John Wiley and Sons, Ltd., 2000.
- [2] Stephen R. Schach, “Object-Oriented and Classical Software Engineering”, McGraw-Hill Companies, Inc., 2002.
- [3] Jessica Keyes, ”Software Engineering Handbook”, CRC Press LLC, 2003.