On the Criteria To Be Used in Decomposing Systems into Modules

David L Parnas, 1972

Modular Programming

- Given: Modularisation is a good idea
- Advantages:
  - Modules can be written without knowledge about the code of other modules
  - Modules can be reassembled and replaced without reassembling the whole system

Expected Benefits

- Shorter development time
- Greater product flexibility
- Improved Comprehensibility

KWIC

- **K**ey **W**ord **I**n **C**ontext

- Used for reference systems

KWIC

- **We see the following modules:**
- We see the following modules:
- see the following modules: We
- the following modules: We see
- modules: Wee see the following
- following modules: We see the

KWIC

- **We see the following modules:**
- **following** modules: We see the
- **modules**: We see the following
- **see** the following modules: We
- **the** following modules: We see
- **We** see the following modules:

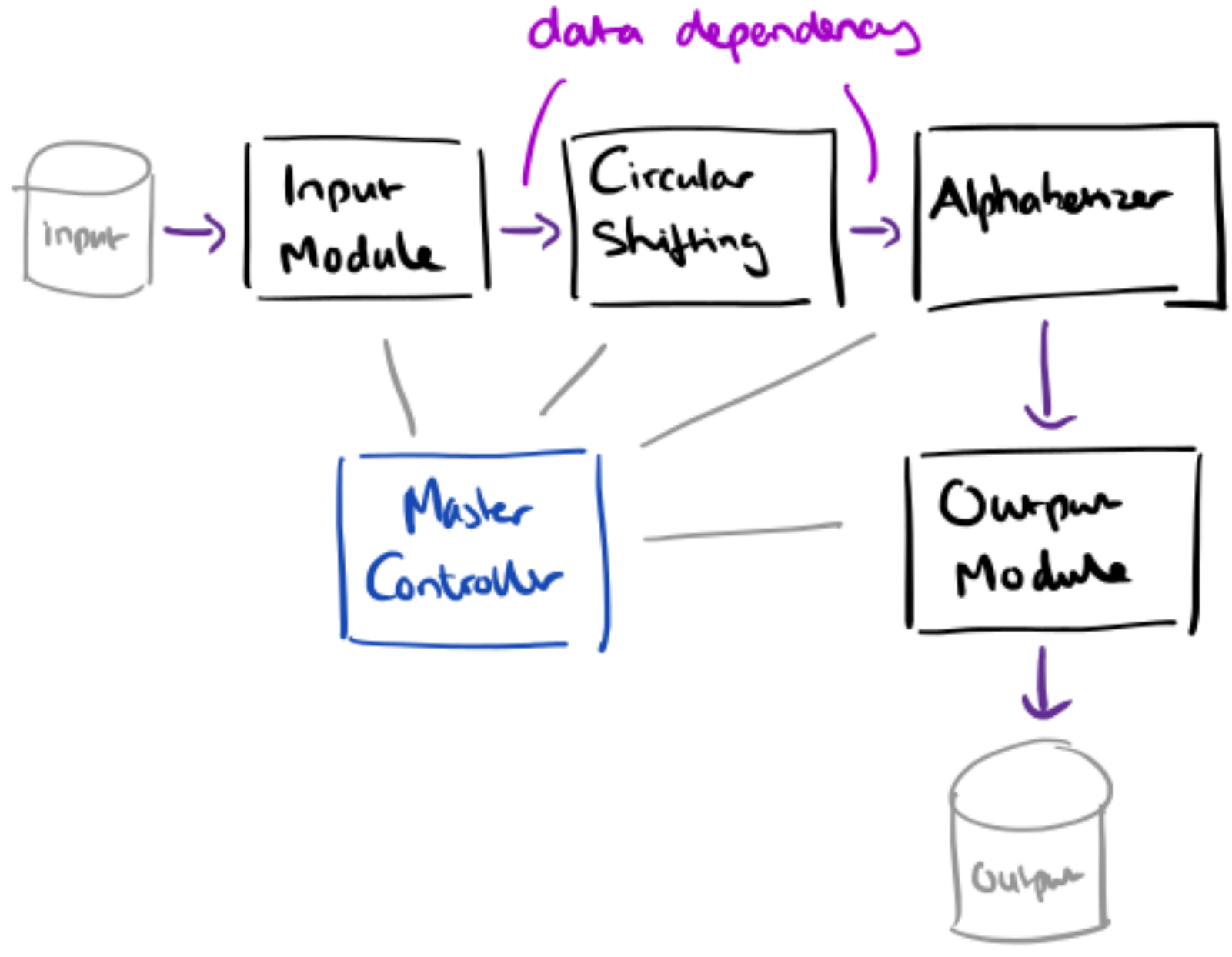
Modularization 1

- **Module 1: Input**
- **Module 2: Circular Shift**
- **Module 3: Alphabetizing**
- **Module 4: Output**
- **Module 5: Master Control**

Modularization 2

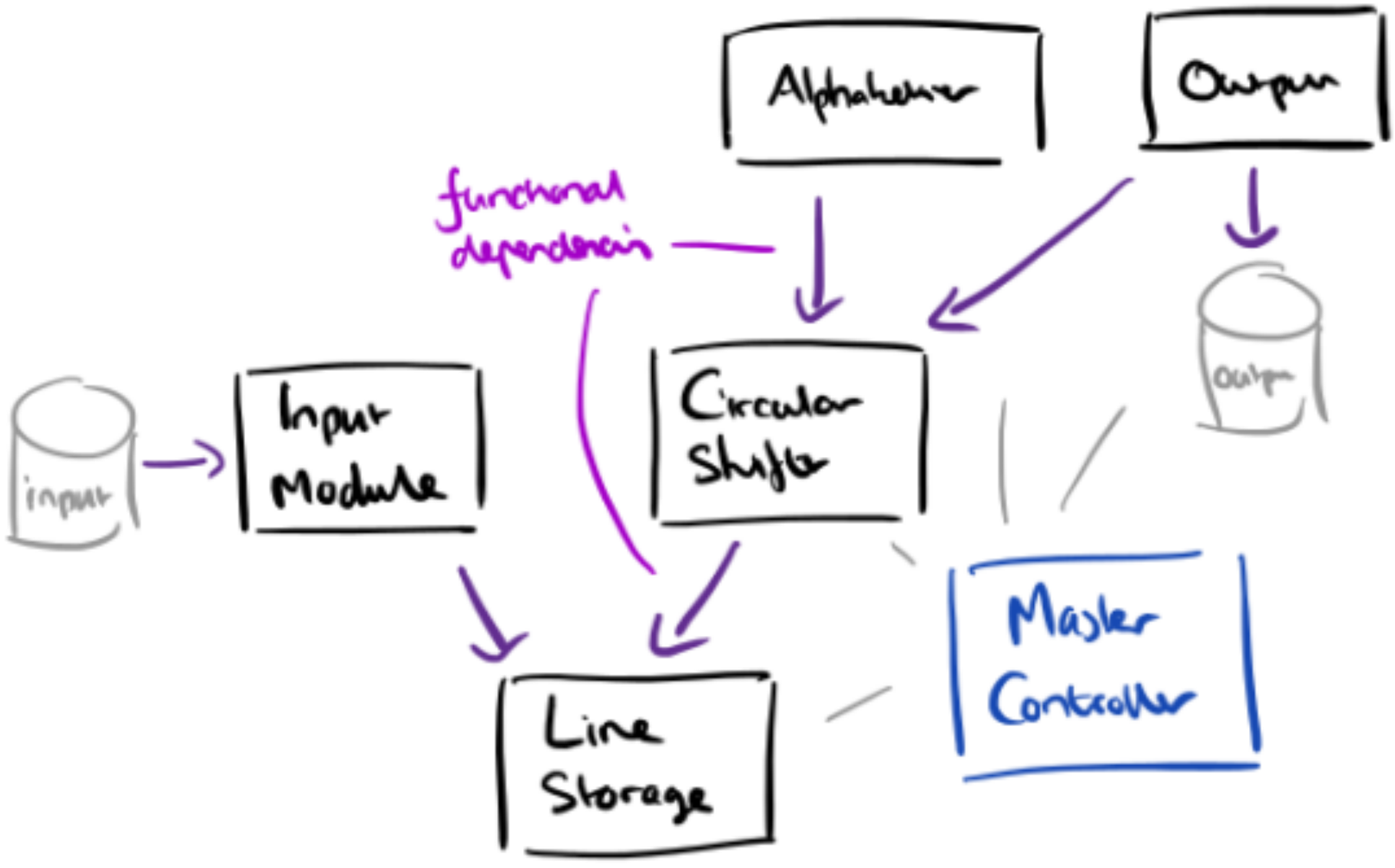
- **Module 1: Line Storage**
- **Module 2: Input**
- **Module 3: Circular Shifter**
- **Module 4: Alphabetizer**
- **Module 5: Output**
- **Module 6: Master Control**

Modularization 1



https://blog.acolyer.org/2016/09/05/on-the-criteria-to-be-used-in-decomposing-systems-into-modules/

Modularization 2



https://blog.acolyer.org/2016/09/05/on-the-criteria-to-be-used-in-decomposing-systems-into-modules/

Changes

1. Input format

2. Decision to store all lines in core

3. Decision to pack characters four to a word

4. Decision to store circular shifts as index instead of writing them out

5. Decision to alphabetise once rather then search or partially alphabetise

Independent Development

- Modularization 1: Interfaces between modules are (fairly complex) format and table organisations

- Modularization 2: Interfaces are function names and parameter descriptions

Comprehensibility

“The system [1] will only be comprehensible as a whole. It is my subjective judgment that this is not true in the second modularization.“

The Criteria

- Modularization 1: Flowchart based

- Modularization 2: Information hiding

Efficiency and Implementation

- Danger: Less efficient because of many switches between modules

- Proposal: Way to write code that *assembles* subroutines to inline them into calling code

Comprehensibility

“We have tried to demonstrate by these examples that it is almost always incorrect to begin the decomposition of a system into modules on the basis of a flowchart. **We propose instead that one begins with a list of difficult design decisions or design decisions which are likely to change**. Each module is then designed to hide such a decision from the others.“

Resources

- https://www.win.tue.nl/~wstomv/edu/2ip30/references/criteria_for_modularization.pdf
- https://prl.ccs.neu.edu/img/p-tr-1971.pdf
- https://blog.acolyer.org/2016/09/05/on-the-criteria-to-be-used-in-decomposing-systems-into-modules/
- https://www.slideshare.net/ufried/excavating-the-knowledge-of-our-ancestors