Structured Design

Structured Design

- **HIPO – H**eirarchical **I**nput **P**rocess **O**utput

  » **Based on the fact that programs have input and output with some processing inbetween**

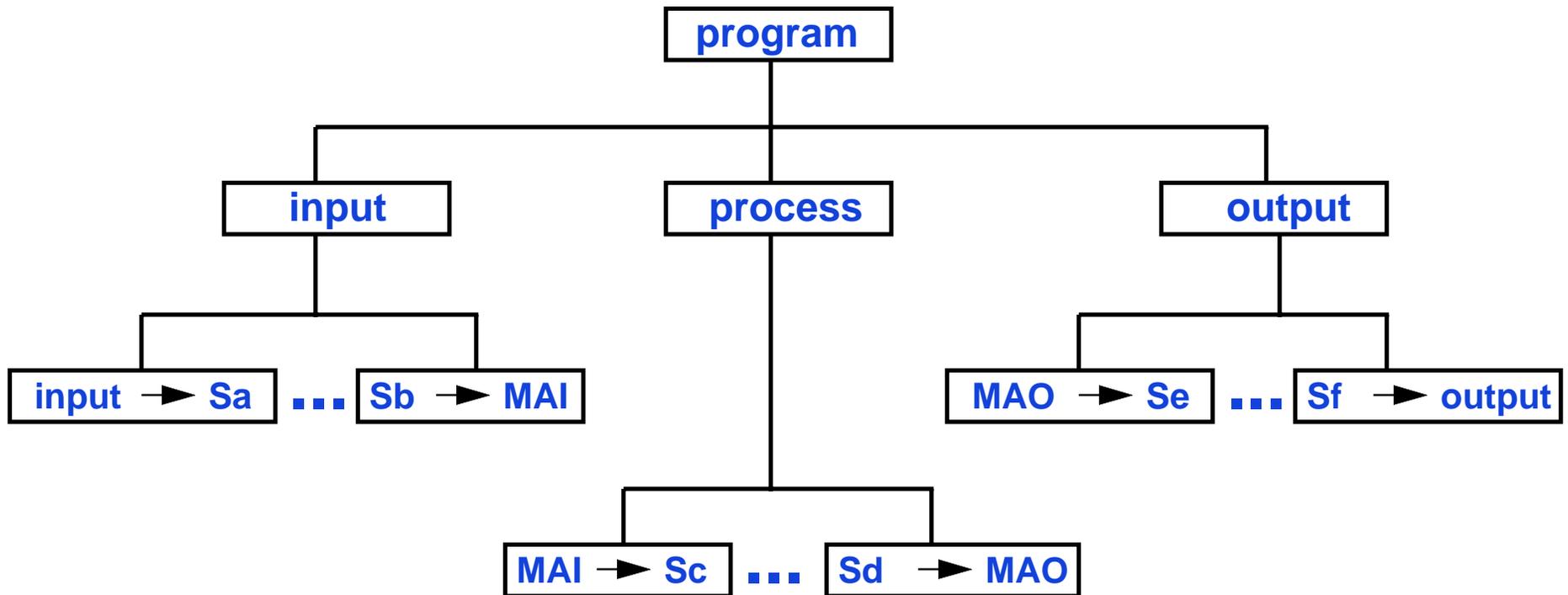- Often can identify a chain of transformations from input to output

| input | → | S1 | → | S2 | → | S3 | → | S4 | → | S5 | → | output |

Structured Design – 2

- In particular can identify two intermediate points

  » **The Most Abstract Input**

    > **The transformation of the external input to its corresponding internal form ready for processing**

  » **The Most Abstract Input**

    > **The transfromation of the internal form after processing to the external output**

input → ... → **MAI** → ... → **MAO** → ... → **output**

Structured Design – 3

- This leads to the following architecture

```
                        program
                           │
        ┌──────────────────┼──────────────────┐
      input             process             output
        │                  │                    │
   ┌────┴────┐             │              ┌──────┴──────┐
 input → Sa ... Sb → MAI   │          MAO → Se ... Sf → output
                      ┌─────┴─────┐
                   MAI → Sc ... Sd → MAO
```

License Plate Problem

Main Program

1 ↑

4 ↓

Read char input & create integer representation

Input →

2 ↓ **3** ↑

Create char representation & write output

→ Output

Compute next plate number

1 Old plate number as integer digits

2 Old plate number as integer digits

3 New plate number as integer digits

4 New plate number as integer digits

© Gunnar Gotshalks