Table of Contents

- Definitions
- Large System Development Costs
- Development Goals
- Poor Communication
- Design Communication
- Design Elements
- Structure Chart
- Parameter Notation
- Parameter Example
- Flow Control: Selection
- Flow Control: Loop
- Misc. Routine Calls
- Misc. Routine Calls (continued)
- Connector Symbol
- Interface Diagram
- Global Data
- Global X-Reference Charts

Definitions

Software Systems
- "An integrated whole composed of diverse, interacting, specialized structures and subfunctions." [IEEE]

Software Engineering
- Disciplined systematic technological activity for producing and maintaining software products by means of a controlled efficient process.

Cost of Errors

Relative Cost

Design Code Test

Development Phases
Large System Development Costs

Cost vs. Number of Modules

System Development Cost Relationships

- Interfacing
- Integration Cost
- Total Cost
- Cost per Component

Number of Modules

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Cost</th>
<th>Instructions (Millions)</th>
<th>Effort (man years)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apollo</td>
<td>$209</td>
<td>23</td>
<td>6000</td>
</tr>
<tr>
<td>Skylab</td>
<td>$ 30</td>
<td>1.25</td>
<td>1,000</td>
</tr>
<tr>
<td>NASA satellite range monitoring</td>
<td>$103</td>
<td>1.48</td>
<td>5,000</td>
</tr>
<tr>
<td>FAA air traffic control Safeguard ABM</td>
<td>$120</td>
<td>1.87</td>
<td>3,500</td>
</tr>
</tbody>
</table>

Development Goals

- Programming
  - Goal: Write Code

  ![Diagram](Problem → Code → Solution)

- Software Design
  - Goals
    - Select components
    - State Functions
    - State Interfaces

  ![Diagram](Problem → Solution)
Poor Communication

Design Communication

- Design Document
  - Concrete and Specific
  - Can be analyzed
  - Can be compared to implementation
  - Traceable through life cycle
Design Elements

- A Design should contain
  - Components
  - Procedures
  - Functions

- Function of Each Component
  - Suggestive names

- Interfaces
  - Control
    - Calling Hierarchy
  - Data
    - Parameters
    - Global Variables
    - Files

Structure Chart

- Structure Chart
  - A graphic tool used to hierarchically model the design solution of a problem.

- Contains:
  - Individual modules
  - Data passed to/from modules
  - Control Interfaces between modules

- Basic Elements

```
Calling Module: GetWord

Called Module: GetNext Char
```

Control Relationship
**Parameter Notation**

- Parameter Direction Flow
  - 3 Types
    - 1. Input: Value Parameters & Const Reference Parameters
    - 2. Output: Reference Parameters (Function changes parameter values independently of parameter’s original [passed] value.)
    - 3. I/O: Reference Parameters (Function changes parameter values dependent upon of parameter’s original [passed] value.)

- Parameter Classes
  - 2 Classes
    - 1. Data Parameter: Any parameter upon which NO decision in the called module’s or calling module’s code is based.
    - 2. Control Parameter: Any parameter upon which a decision in the called module’s or calling module’s code is based.

**Parameter Example**

```c
void GetNextChar
...
GetChar (ch, class, terminator);
if (terminator) ...
```

```
Get Next Char

class <- ch

Get Char

void GetChar
(char& chact ,
 uplowtype clas,
 bool& terminal )
```
Flow Control: Selection

What does the function header for `DrawBox` look like?

```c
void Move
...
    if ( ch == plus )
        DrawBox (row, col);
...
```

Flow Control: Loop

```c
void Drawbox
...
    while (ch != plus)
    {
        ... 
        DrawLine ( dir );
        ...
    }
...
```
- Recursive Routines
  - Routines that call themselves

- Operating System Calls

  - Parallel Processing
    - Methods
    - FORK / JOIN
    - PARBEGIN / PAREND

  - Predefined Module (library routines)

```
PARBEGIN
  ProcessA (...);
  ProcessB (...);
PAREND
```
Large designs span many pages

Interface Diagram #3

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Dir</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>data</td>
<td>in</td>
<td>cartesian coords</td>
</tr>
<tr>
<td>y</td>
<td>data</td>
<td>in</td>
<td>of point</td>
</tr>
<tr>
<td>z</td>
<td>data</td>
<td>in</td>
<td></td>
</tr>
<tr>
<td>scale</td>
<td>data</td>
<td>in</td>
<td>scale of axis</td>
</tr>
<tr>
<td>angrad</td>
<td>cont</td>
<td>in</td>
<td>angle degrees (rad)</td>
</tr>
<tr>
<td>angle</td>
<td>data</td>
<td>out</td>
<td>polar coords</td>
</tr>
<tr>
<td>dist</td>
<td>data</td>
<td>out</td>
<td>of point</td>
</tr>
</tbody>
</table>
Global Data

- Global Data (extern)
  - Must be highly justifiable
  - Operation
    - Assign
    - Update
    - Reference

Global X-Reference Charts

- Global Variables introduce complexities that must be mapped and traceable.

Global Structures

<table>
<thead>
<tr>
<th>Procedures</th>
<th>Form</th>
<th>Setting</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetNext</td>
<td>R</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>Advance</td>
<td>U</td>
<td>A</td>
<td>R</td>
</tr>
<tr>
<td>PutNext</td>
<td>A</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>Reset</td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
</tbody>
</table>

U : Update  A : Assign  R : Reference