IMPLEMENTATION SUPPORT

Objectives:
- Programming Tools for interactive systems provide:
  - a means of effectively translating abstract designs and usability principles into an executable form
  - different level of services for the programmer

- Windowing Systems:
  - central environment for the programmer and the user
  - allowing a single workstation to support separate user-system threads of action simultaneously

- Interaction Toolkits:
  - allowing the programmer to describe behaviors of objects at a level similar to how the user perceives them

- User Interface Management Systems (UIMS):
  - the final level of programming support tools
  - allowing the designer and the programmer to control the relationship between the presentation objects of a toolkit with their functional semantics in the actual application

Elements of Windowing Systems

Two features:
- Device Independence ➔ konsep abstract terminal
- Resource Sharing ➔ multiple application control

Abstract Terminal:
- makes the programming task easier
- makes portability of application programs possible

Device Driver (translation program):
- needs to be written for a particular hardware device and then any application program can access it.

Contoh Abstract Terminal:
- Pixels
- Graphical Kernel System (GKS)
- Programmer's Hierarchical Interface to Graphics (PHIGS)
- PostScript
Fig. 01. The Roles of Windowing System
Architectures of Windowing Systems

Clients

Client Application 1

Client Application 2

... Client Application n

Abstract Terminal 1

Abstract Terminal 2

... Abstract Terminal n

Server

Resource Manager

Device Driver

Devices

Mouse

Win. 1

Win. 2

Win. n

Keyboard

Fig. 02. The Client Server Architecture
Programming The Application

- Programming Paradigm:
  - Read-Evaluation Loop
  - Notification Based

![Flowchart of the Read-Evaluation Loop Paradigm](image)

**Fig. 03. The Read-Evaluation Loop Paradigm**
Fig. 04. The Notification Based Programming Paradigm
User Interface Management Systems (UIMS)

The main concerns of a UIMS:

- A conceptual architecture for the structure of an interactive system which concentrates on a separation between application semantics and presentation
- Techniques for implementing a separated application and presentation
- Support techniques for managing, implementing and evaluating a run time interaction environment

The Logical Components of a UIMS:

- Presentation
- Dialogue control
- Application interface

![Diagram showing the logical components of a UIMS](image)

Fig. 05. The Seeheim model of the logical components of a UIMS
Fig. 06. The Model-View-Controller triad in Smalltalk

Fig. 07. The Presentation-Abstraction-Control model of Coutaz