

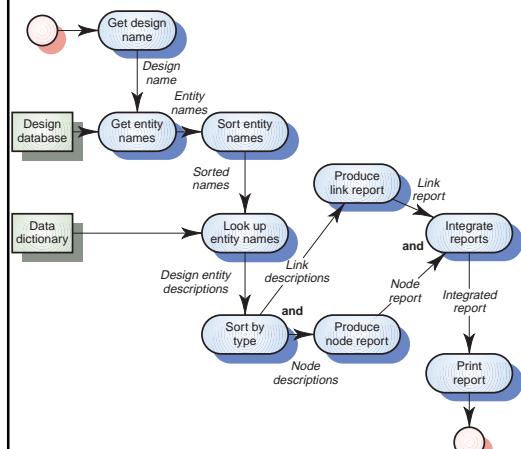
ATM software design

```
loop
  loop
    Print_input_message (" Welcome - Please enter your card");
    exit when Card_input;
  end loop;
  Account_number := Read_card;
  Get_account_details (PIN, Account_balance, Cash_available);
  if Validate_card (PIN) then
    loop
      Print_operation_select_message ;
      case Get_button is
        when Cash_only =>
          Dispense_cash (Cash_available, Amount_dispensed);
        when Print_balance =>
          Print_customer_balance (Account_balance);
        when Statement =>
          Order_statement (Account_number);
        when Check_book =>
          Order_checkbook (Account_number);
      end case;
      Eject_card;
      Print ("Please take your card or press CONTINUE");
      exit when Card_removed;
    end loop;
    Update_account_information (Account_number, Amount_dispensed);
  else
    Retain_card;
  end if;
end loop;
```

©Ian Sommerville 1995 Software Engineering, 5th edition, Chapter 15

Slide 7

Design report generator



©Ian Sommerville 1995 Software Engineering, 5th edition, Chapter 15

Slide 12

OIRS design

```
procedure OIRS is
begin
    User := Login_user ;
    Workspace := Create_user_workspace (User) ;
    -- Get the users own document database using the user id
    DB_id := Open_document_database (User) ;
    -- get the user's personal index list;
    Known_indexes := Get_document_indexes (User) ;
    Current_indexes := NULL ;
    -- command fetch and execute loop
    loop
        Command := Get_command ;
        exit when Command = Quit ;
        Execute_Command (DB_id, Workspace, Command, Status)
        if Status = Successful then
            Write_success_message ;
        else
            Write_error_message (Command, Status) ;
        end if ;
    end loop ;
    Close_database (DB_id) ;
    Logout (User) ;
end OIRS ;
```

©Ian Sommerville 1995 Software Engineering, 5th edition, Chapter 15

Slide 31

Detailed process design

```
task body Get_command is
begin
    -- Fetch/execute loop
    loop
        loop
            Cursor_position := Get_cursor_position ;
            exit when cursor positioned in workspace or
                (cursor positioned over menu and button pressed)
            Display_cursor_position ;
        end loop ;
        if In_workspace (Cursor_position) then
            Workspace_editor.Enter ;
        elsif In_Command_menu (Cursor_position) then
            Process_Command.Command_menu ;
        elsif In_Known_Indexes (Cursor_position) then
            Process_Command.Display_Indexes ;
        elsif In_Current_Indexes (Cursor_position) then
            ...
            Other commands here
            ...
        end loop ; -- Fetch/execute
    end Get_Command ;
    -- other task implementations here
end Office_system ;
```

©Ian Sommerville 1995 Software Engineering, 5th edition, Chapter 15

Slide 35