

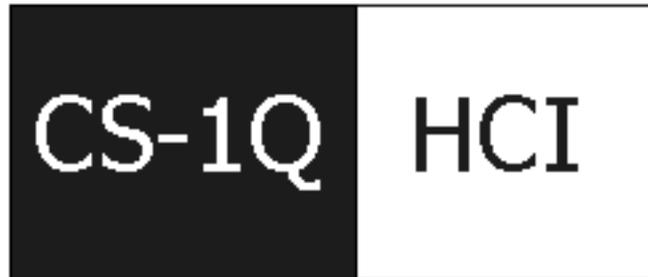
# Desktop Interaction

C.W. Johnson,

University of Glasgow,  
Glasgow, G12 8QQ.  
Scotland.

johnson@dcs.gla.ac.uk,  
<http://www.dcs.gla.ac.uk/~johnson>

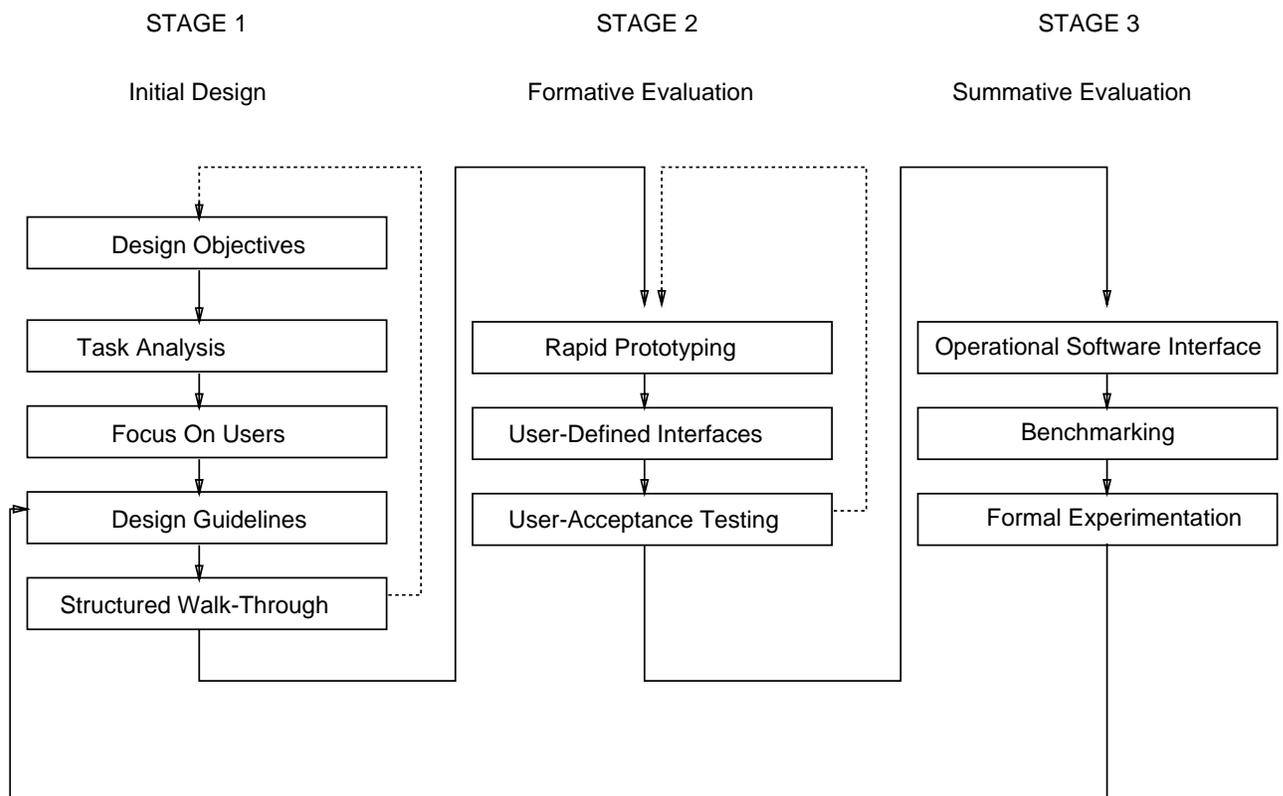
October 2001



---

# Desktop Interaction

- How do we implement a desktop system?



- Iterative development and user-centred design

---

# Desktop Interaction

1. consider users and environment;
2. design and specification;
3. select dialogue style;
4. implementation issues;
5. documentation issues;
6. evaluation issues.

---

# 1. Consider Users and Environment

- Home:
  - cluttered with books, coffee cups etc;
  - distractions from radio, other people etc.



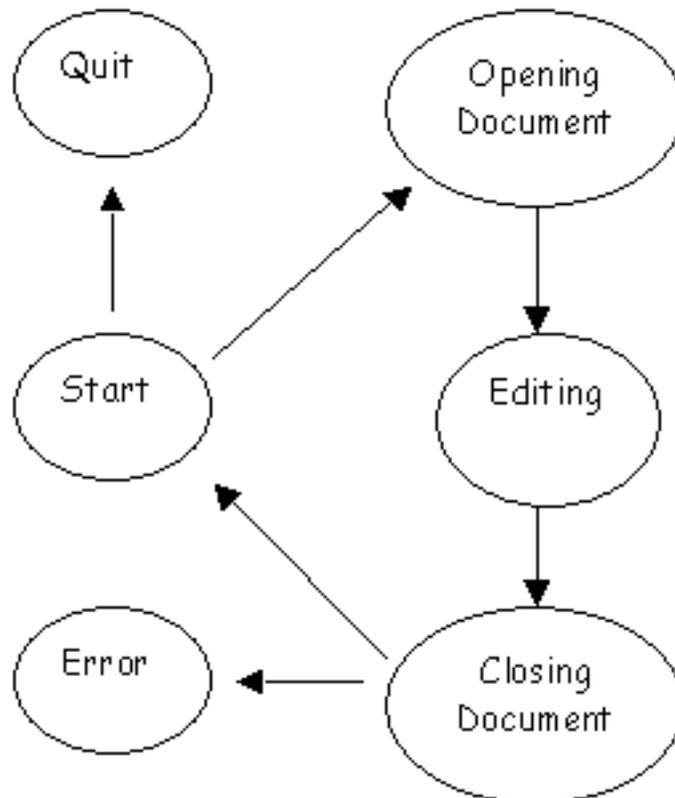
Acknowledgement: BBC

- Office:
  - time pressure to complete tasks;
  - need to achieve *closure*;
  - heat, noise, interruptions, multiple tasks.

---

## 2. Design and Specification

- Task analysis looks from users' point of view.
- Specifications begin to look at system behaviour:
  - focus on *what* to do not *how* to do it.



- State-transition diagrams often used.

---

## 2. Design and Specification

- States:
  - situations where system continues to perform activity;
  - change from a state is triggered by a transition;
  - represented by nodes in the network.
  
- Transitions:
  - edges represent change between states;
  - usually triggered by user actions;
  - can be labelled by frequency information.
  
- Can do static analysis:
  - can you get from state A to B (reachability);
  - maximum of N transitions from A to D (spanning).

---

## 3. Select Dialogue Style

- Dialogue styles:
  - text, forms, menus, graphics, mixtures...



- Dialogue styles rely on widgets:
  - lists, choices and buttons;
  - checkboxes and radio buttons;
  - pull-down, tear-off, pop-up, scrolling, hybrid menus

---

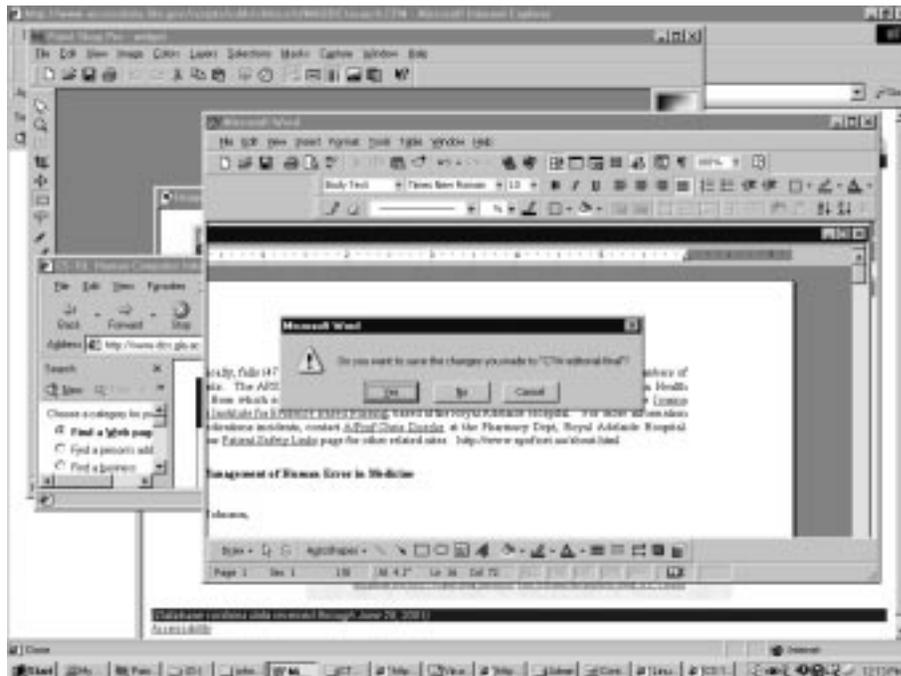
## 3. Select Dialogue Style

- Windows:
  - titles, borders and frames, scroll bars.
  
- Windowing actions:
  - open, move, close, resize, select, bring forward,
  - feedback is crucial to all of these actions.
  
- Some very complex web-behaviours:
  - spontaneously opening a linked window with a web page;
  - 'where on earth did that come from?'
  - lots of security issues here - unsigned applets.

---

### 3. Select Dialogue Style

- Window managers between applications.
- Layout managers within an application.



- Placement issues:
  - multiple overlapping windows (how to select a window?);
  - tiling strategies (eg powerpoint slide view);
  - zooming (eg Acrobat page size);
  - card cascades etc.

---

## 4. Implementation Issues

- How do you get a widget in your code?

- Toolkits such as AWT, Swing (or JEWL).

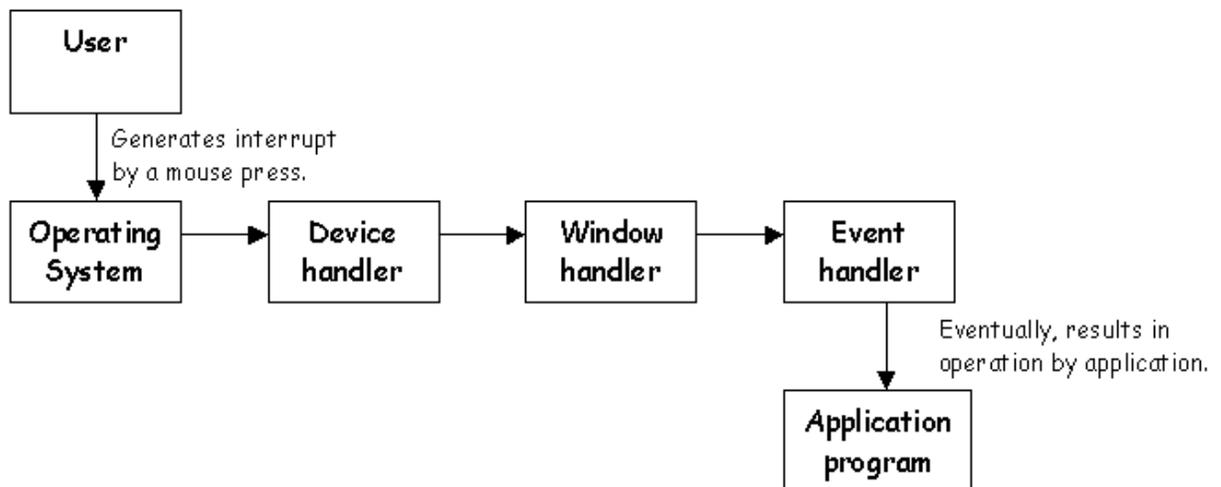
```
/* do NOT memorise this code! */  
/* Try to understand what it is doing */  
b1 = new Button(); // create a new button  
b1.setLabel("Disable middle button"); // put a piece of text in it  
b1.setActionCommand(DISABLE); // associate a command with it
```

- Toolkits are libraries of programs:
  - others have written them to implement common widgets;
  - you don't have to write code to draw a button etc;
  - can be difficult to change these pre-coded widgets.

---

## 4. Implementation Issues

- What happens when button is clicked?



```
public void actionPerformed(ActionEvent e) {  
    String command = e.getActionCommand();  
    if (command == DISABLE)  
        {} // b1 was pressed so do whatever you need to  
}
```

---

## 4. Implementation Issues

- User interface programming is skilled:
  - can take 3-4 months to learn the basics.
  
- Lots of tools to simplify the process.
  
- Direct manipulation interface builder:
  - select widget and place it on 'screen';
  - system automatically generates code you need;
  - can be inflexible and inefficient.
  
- Microsoft Visual Basic, Borland Delphi.
  
- Also scripting tools such as Tcl/Tk.

---

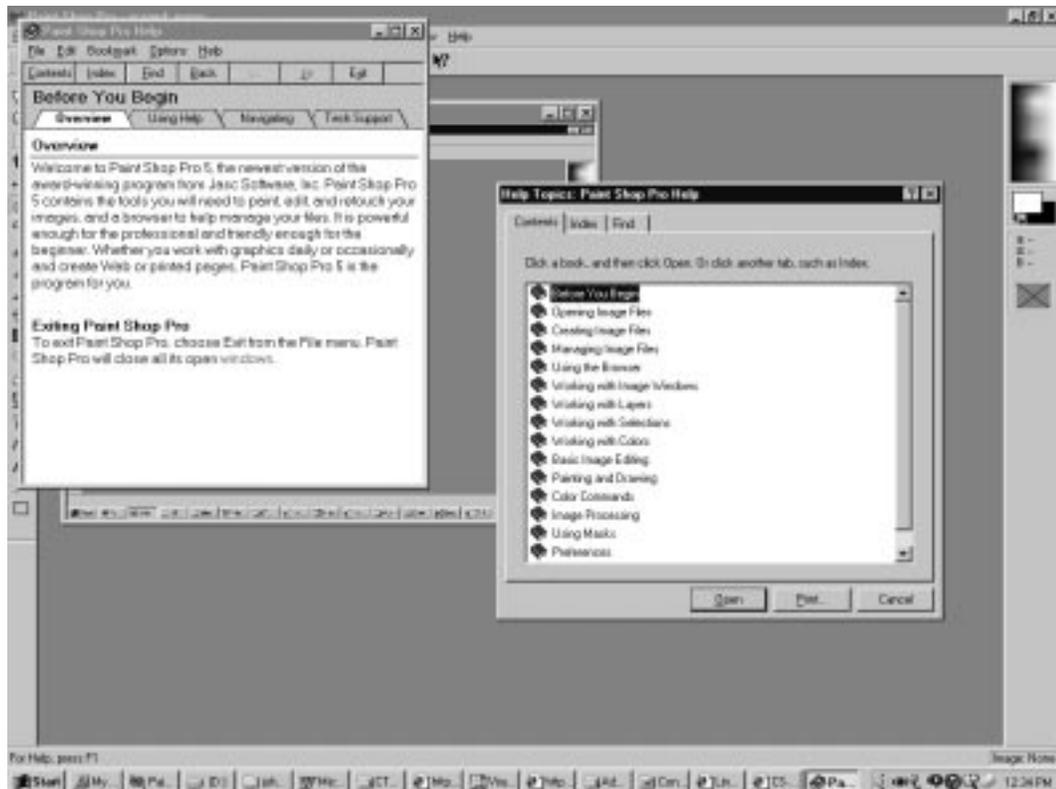
## 5. Documentation Issues

- Printed manuals:
  - seldom read, intimidating and go missing;
  - must stay open at correct page while typing;
  - must leave room for the keyboard and mouse!
  
- Key idea of *minimal manuals*:
  - short 'get you started' summaries;
  - lists of available commands and short cuts.

---

## 5. Documentation Issues

- On-line tutorials and manuals:
  - can be quite basic (Unix apropos/man)

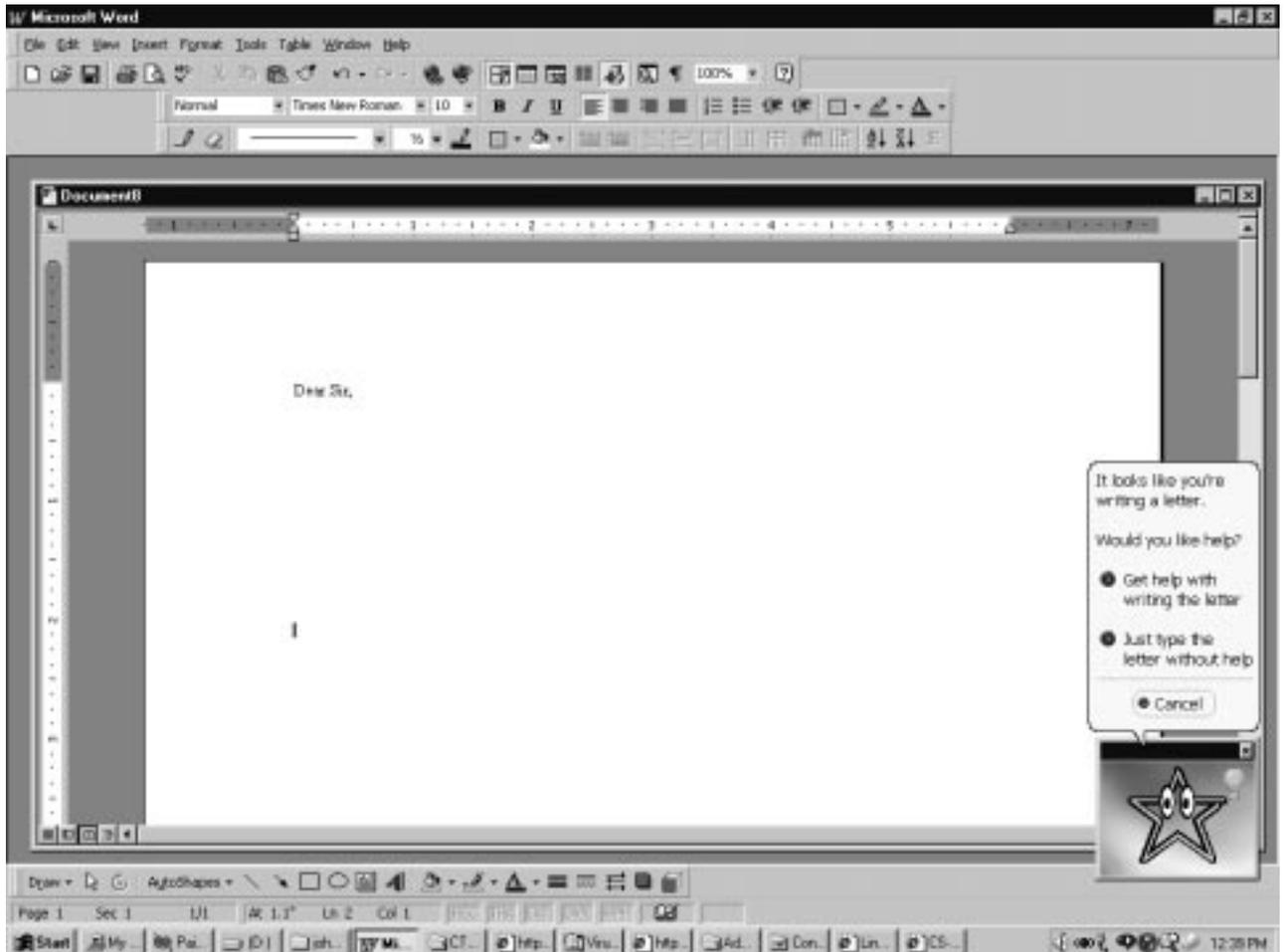


- On-line and context-sensitive help:
  - can be very difficult to navigate;
  - can be out of synch with software installation;
  - can be pitched at too simple/complex a level.

---

## 5. Documentation Issues

- But screen reading is tiring and error prone.



- The infamous 'Office Assistant':
  - does anthropomorphism always support users?
- 'Tip of the day' - can be patronising.

---

## 6. Evaluation Issues

- Recap lecture 4:
  - what is formative evaluation?
  - what is summative evaluation?



- Recap lecture 4:
  - name one formative technique?
  - name one summative technique?

---

# Summary

1. consider users and environment;
2. design and specification;
3. select dialogue style;
4. implementation issues;
5. documentation issues;
6. evaluation issues.

---

# Further Reading

- Shneiderman on:  
- software tools - pp. 155-179.