Khulna University
Computer Science & Engineering Discipline

SK Alamgir Hossain

Assistant Professor
Computer Science & Engineering Discipline
Khulna University, Khulna, Bangladesh

http://alamgirhossain.com

CSE3203 - Software Engineering and Information System Design

Chapter 10

– Designing Human Interface

Chapter Outline

Khulna University

- Designing Human Interface
  - User Interface
  - Human factors
  - Types of User Interface
  - Guidelines for user interface design
  - Database Design: Normalization

Lecture 19

Designing Human Interface

- User Interface
- Human factors
- Types of User Interface
- Guidelines for user interface design
- Database Design: Normalization

What is a User Interface?

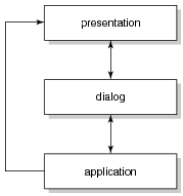
- The user interface, or the human/computer interface is what the user sees, and includes:
- the physical controls – buttons, etc.
- what the system looks like (if there is a monitor
  – the system could be a washing machine or a photocopier)
- how the system accepts input from the user
- how the system responds to user input
- how the system outputs the results of processing

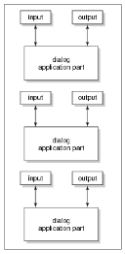
What is the user interface?

- User interface: all aspects of a system that are relevant to the user
- Also called: User Virtual Machine (UVM)
- A system can have more than one UVM, one for each set of tasks or roles
- An individual may also have more than one user interface to the same application, e.g. on a mobile phone and a laptop

Where is the user interface?

- Seeheim model: separate presentation and dialog from application
- More recently: MVC – Model-View-Controller





Specialised User Interfaces

- There are many examples of computer applications with specialised interfaces:
- the tills in pubs, MacDonalds, etc., are often just ordinary PCs with specialised keyboards
- games consoles – the Super Nintendo was a 6502-based machine, like the BBC Model B
- computers are often adapted for people with disabilities – e.g. computers operated by blow-pipes, and Stephen Hawking's speaking computer

What Makes a Good User Interface?

- Good user interfaces are:
- Safe – not ambiguous or confusing (especially in safety critical applications)
- Effective – they do what they're supposed to do, and quickly
- Efficient – they are clear and easy to use
- User-friendly – intuitive and easy to learn
- Enjoyable – HCIs can be irritating, e.g. "Are you sure?" messages with no Yes or No buttons

What Makes a Good User Interface?

- Sometimes there might be a simple solution:
- When ATMs (cash machines) were first introduced, people kept leaving their cards in them.
- How did banks solve this problem?
- They just made the machines beep until the card was removed!

Two ways to look at a user interface

- Design aspect: how to design everything relevant to the user?
- Human aspect: what does the user need to understand?

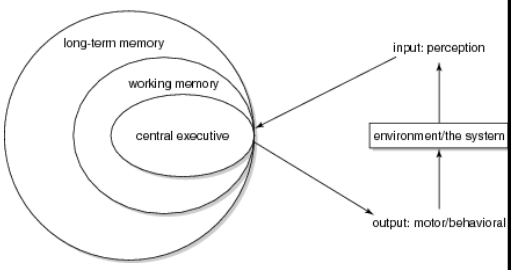
Human factors

- Humanities
  - Psychology: how does one perceive, learn, remember, …
  - Organization and culture: how do people work together, …
- Artistic design
  - Graphical arts: how doe shapes, color, etc affect the viewer
  - Cinematography: which movements induce certain reactions
  - Getting attractive solutions
- Ergonomics
  - Relation between human characteristics and artifacts
  - Especially cognitive ergonomics

Models in HCI

- Internal models ('models for execution')
  - Mental model (model of a system held by a user)
  - User model (model of user held by a system)
- External models ('for communication')
  - Model of human information processing
  - Conceptual models (such as Task Action Grammar)

Model of human information processing



Use of mental models

- Planning the use of technology
  - First search by author name
- Finetuning user actions while executing a task
  - Refine search in case of too many hits
- Evaluate results
  - Keep the titles on software engineering
- Cope with events while using the system
  - Accept slow response time in the morning

Characteristics of mental models (Norman)

- They are incomplete
- They can only partly be 'run'
- They are unstable
- They have vague boundaries
- They are parsimonious
- They have characteristics of superstition

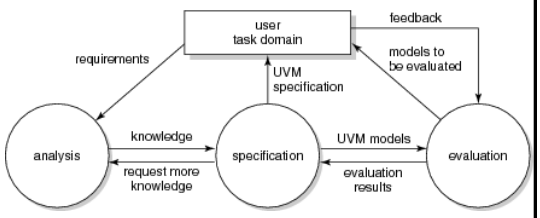
Conceptual model

- All that is modeled as far as it is relevant to the user
- Formal models
  - Some model the user's knowledge (competence model)
  - Others focus on the interaction process
  - Others do both

Viewpoints of conceptual models

- Psychological view: definition of all the user should know and understand about the system
- Linguistic view: definition of the dialog between the user and the system
- Design view: all that needs to be decided upon from the point of view of user interface design

Design of the user interface



Designing the User Interface

- When designing the HCI, what will you consider?
- Who will be using the system:
  - What is their level of skill and knowledge?
  - Are there any special physical requirements?
  - Will there be a range of users?
- What does the system actually do? How many options are there, and how accessible do they need to be?
- The environment in which it will be used – e.g. military use, temperature, noise, motion, moisture
- The technology available – iris recognition, voice, etc.

Types of User Interface

- There are six types of UI that you need to know:
- Command line – e.g. DOS or SQL
- Menu-driven – e.g. old DOS applications
- Natural language – beloved of science fiction!
- Forms and dialogue boxes – used in Windows applications – e.g. Print or Browse
- Graphical User Interface – also known as GUIs, or WIMP (window, icon, menu, pointer, or window, icon, mouse, pull-down menu)

Command Line Interfaces

- Often very flexible
- Good for expert users who know the commands – not hunting through menus for the option you require
- Usually require fewest computer resources

- Requires users to learn commands
- Things you don't know about are invisible
- Not very good for novices or people who can't type

Menu-Driven Interfaces

- All of the options are visible – no hidden features
- No obscure commands and syntax to learn – therefore the it will be easier for new users and less training will be required

- Can be frustrating for experienced users who want to do something buried behind 10 menus!
- The user interface may be limited – e.g. by the size of the screen and the number of options you can fit on

Natural Language Interfaces

- No training required – you just tell it what you want!
- They could potentially be as powerful and flexible as CLIs
- No menus to negotiate
- Makes the computer look intelligent to naïve users

- Some sentences are ambiguous – "Let him have it!"
- It would be difficult to recognise all the different ways of saying things (and regional dialects)
- Artificial languages are often more precise
- Not yet popular

Forms and Dialogue Boxes

- Issues when designing forms and dialogue boxes:
- Navigation – order of fields, tabbing, etc.
- Layout – not too cluttered
- Field sizes appropriate to data expected
- Use of appropriate controls and validation – combo-boxes, radio buttons, checkboxes, etc.
- Whether forms and dialogue boxes are *modal* – i.e. can the user access anything else while they are displayed?

Use of Sound

- Feedback using sound is easily noticed – e.g.
  - Telephones
  - Alarm clocks
  - E-mails arriving in Outlook!
- Users can find too much noise distressing, so it needs to be used carefully
- Sounds won't necessarily be heard in a noisy environment such as a factory
- Sound output could be in the form of speech – either synthesised or digitised

Speech Recognition

- Speech can also be used for input:
- Command and Control – small vocabulary systems can be used for controlling devices or systems. They are usually better at recognising different voices as there are fewer words to differentiate – e.g. just Yes or No
- Dictation systems – e.g. Office or ViaVoice, these are large vocabulary systems used to enter text. They require training for your voice, and tend to be less reliable and more resource hungry than voice control systems.

Common User Interfaces

- Many applications, especially those that run in an operating system with a GUI have a similar appearance – a common user interface.
- Windows applications, for example, all support:
  - F1 to access help
  - Ctrl C for copy, X for cut and V for paste
  - Use of the Tab key to move between controls
  - Use of the Alt key to operate menus
  - Use of the spacebar to depress a button
  - Use of Ctrl and click to select multiple items
  - Plus many more!

Advantages of a Common Interface

- Having a common user interface brings benefits:
- It's quicker to learn new applications
- Familiar interfaces make applications easier to use
- All applications looking the same makes inexperienced users more confident
- Once an ICT "expert" is familiar with Windows, they should be able to operate almost any application

Guidelines for user interface design

- Use a simple and natural dialog
- Speak the user's language
- Minimize memory load
- Be consistent
- Provide feedback
- Provide clearly marked exits
- Provide shortcut
- Give good error messages

Summary

- Central issue: tune user's mental model (model in memory) with the conceptual model (model created by designers)
- User interface design requires input from different disciplines: cognitive psychology, ethnography, arts, …

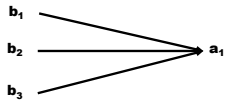
Lecture 20

Designing Human Interface

- User Interface
- Human factors
- Types of User Interface
- Guidelines for user interface design
- Database Design: Normalization

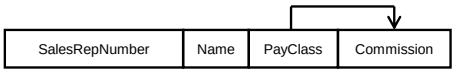
Functional Dependence

- An attribute A is functionally dependent on attribute(s) B if: given a value b for B there is one and only one corresponding value a for A (at a time).



Example: functional dependence

- All sales representatives in a given pay class have the same commission rate.



Keys

- Primary Key: a minimal set of attributes that form a candidate key
- Any attribute or collection of attributes that functionally determine all attributes in a record is a Candidate Key.
- Note: since no two rows in a relational table can be duplicates, the entire record is always a candidate key.

Primary Key (C)

- C determines all attributes
- No subset of the attributes in C is a candidate key
- A key consisting of more than one attribute is called a “composite key.”

Good Primary Keys

- Do not change over the life of the database
- Are not "intelligent keys"
- Are not too long
- Do not consist of too many attributes (3 or fewer is good)

Foreign Keys

- A value in the "child" table that matches with the related value in the "parent" table.
- SalesRep(SalesRepNumber, Name)

  [03 | Mary Jones]

  [124 | 03]
- Customer(CustomerNumber, SalesRepNumber)

Foreign Keys

- The foreign key in the child table has the same value as the primary key in the parent.
- The foreign key in a many-to-many relationship goes in the many table.
- In a many-to-many relationship, foreign keys from both tables go into an associative entity.
- In a 1-to-1 relationship the foreign key goes into one of the tables (usually the one most likely to change)

Normal Forms

- A set of conditions on table structure that improves maintenance.
- Normalization removes processing anomalies:
  - Update
  - Inconsistent Data
  - Addition
  - Deletion

Normal Forms

- All attributes depend on the key, the whole key and nothing but the key.
- 1NF Keys and no repeating groups
- 2NF No partial dependencies
- 3NF All determinants are candidate keys
- 4NF No multivalued dependencies

1st Normal Form

- Table has a primary key
- Table has no repeating groups

- A multivalued attribute is an attribute that may have several values for one record
- A repeating group is a set of one or more multivalued attributes that are related

Example

- Multivalued attribute:

  Orders(OrderNumber, OrderDate, {PartNumber})

  [12491 | 9/02/2001 | BT04, BZ66]

- Repeating group:

  Orders(OrderNumber, OrderDate, {PartNumber, NumberOrdered})

  [12491 | 9/02/2001 | (BT04, 1), (BZ66, 1)]

Normalization: 1NF

- Every repeating group becomes a new table with the appropriate foreign key relationships preserved.
- Remove nested repeating groups from the outside in
- Order(OrderNumber, OrderDate, {PartNumber, {Supplier}})

Example: 1NF

- Order(OrderNumber, OrderDate, {PartNumber, {Supplier}})
- Order(OrderNumber, OrderDate)
- Order-Part(OrderNumber, PartNumber)
- Part(PartNumber, {Supplier})
-

Example: 1NF (cont.)

- Part(PartNumber, {Supplier})
- Part(PartNumber)
- Part-Supplier(PartNumber, SupplierNum)
- Supplier(SupplierNum)

2nd Normal Form

- No partial dependencies
- No attribute depends on only some of the attributes of a concatenated key.
- Order-Part
- [OrderNumber | PartNumber | PartDescription]
- Create a new table with PartNumber key.

3rd Normal Form / Boyce-Codd Normal Form

- 3rd Normal Form: no transitive dependencies
- Transitive dependency means that a non-key attribute depends on another non-key attribute(s).
- This definition says nothing about dependencies that involve the key.

3rd Normal Form / Boyce-Codd Normal Form

- BCNF: every determinant is a candidate key.
- Determinant: any attribute(s) that functionally determine another attribute
- BCNF means that there are no "transitive" dependencies involving key or non-key attributes.

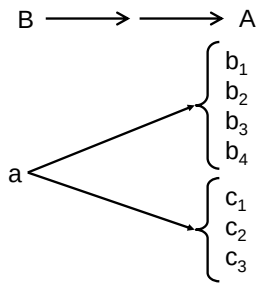
3NF

- Pratt and Adamski use the BCNF definition as their definition of 3NF
- BCNF was generated to deal with problems like:
- Class(Section#, InstructorID, …) extra key attribute
- (Student, Major, Advisor) wrong key

4th Normal Form

- No multivalued dependencies
- A multivalued dependency of column B on column A occurs when a table has a key with three or more attributes, (A, B, C) and
  - each value of A is associated with a collection of values of B
  - this collection of values is independent of C

B is multidependent on A



Example: multivalued dependence

- Faculty have advisee assignments for one major and committee assignments
- [FacID | AdviseeID | FacultyComm]

[FacultyID]

- [FacID | AdviseeID] [FacID | FacultyComm]

Normalization

- Improves maintenance for database changes
- Tends to slow down retrieval
- Better at finding problems than solving them
- Standard normalization procedures are subtle and may introduce BCNF or 4NF problems into tables

Normalization

- 1NF Keys & no repeating groups
- 2NF 1NF & all attributes depend on all key components
- 3NF 2NF & all determinants are candidate keys
- 4NF 3NF & no multivalued dependencies

Intuitive Normalization

- 1NF Tables represent entities
- 2NF Each table represents only one entity
- 3NF Tables do not contain attributes from embedded entities
- 4NF Triple relationships should not represent a pair of dual relationships

THANK YOU