Computer Systems & Application

Computer System and Application Development Process

By
Mr. Nor Azhar Ahmad
Faculty of Computer Systems & Software Engineering
nazhar@ump.edu.my

Chapter Description

- Aims
  - Able to select suitable components in order to develop a system, and point out the problems statements and translate them into programming codes (flow chart).
- Expected Outcomes
  - Learner’s are able to plan their application development easily
  - Learner’s can implement a rules, associate with the software development
- Other related Information
  - Software Project Management Tutorial
  - Working with project planning
- References
  - Dimri, S. C. 2004. Graphical user interface. New Delhi: Saloni Pub. House.
  - Software Project Management. 2015. S.l.: Pearson Education India.

Content

- Introduction
- Terminologies
- Graphical User Interface Standard (10 Golden Rules of GUI)
  - Visibility of system status
  - Match between system and the real world
  - User control and freedom
  - Consistency and standards
  - Error prevention
  - Recognition rather than recall
  - Flexibility and efficiency of use
  - Aesthetic and minimalist design
  - Help users recognize, diagnose, and recover from errors
  - Help and documentation
- Conclusion

Introduction

- Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements.
- Systems design could be seen as the application of systems theory to product development.

Terminologies

- Computer architecture
- Computer components
- System modules
- System interfaces
- System data

GUI Standard (10 Golden Rules of GUI)

- Rule of thumbs in designing software
- Carefully researched based on human behavior

1. Visibility of system status

- The system should always keep users informed about what is going on, through appropriate feedback within reasonable time

2. Match between system and the real world

- The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms.
- Follow real-world conventions, making information appear in a natural and logical order.

3. User control and freedom

- Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue.
- Support undo and redo.

4. Consistency and standards

- Users should not have to wonder whether different words, situations, or actions mean the same thing.
- Follow platform conventions.
- Results from usability testing:
  - success rate of 80% when people used the navigation scheme structured according to most users' mental model
  - success rate of 9% when people used the navigation scheme structured according to the company's internal thinking

5. Error prevention

- Even better than good error messages is a careful design which prevents a problem from occurring in the first place.
- Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

6. Recognition rather than recall

- Minimize the user's memory load by making objects, actions, and options visible.
- The user should not have to remember information from one part of the dialogue to another.
- Instructions for use of the system should be visible or easily retrievable whenever appropriate.

7. Flexibility and efficiency of use

- Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users.
- Allow users to tailor frequent actions.

8. Aesthetic and minimalist design

- Dialogues should not contain information which is irrelevant or rarely needed.
- Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

9. Help users recognize, diagnose, and recover from errors

- Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

10. Help and documentation

- Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation.
- Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Conclusion

- The process of application development ease the developer and its predecessor managing it
- Graphical User Interface should follow Golden Rules to increase the user adoption

Author Information

NOR AZHAR BIN AHMAD

University Lecturer

Faculty of Computer Systems & Software Engineering

Universiti Malaysia Pahang

Master in Computer Science