

DIALOG DESAIN

Dialog dalam arti umum adalah percakapan antara dua kelompok atau lebih. Sedangkan dialog dalam konteks perencanaan user interface adalah struktur dari percakapan antara user dan sistem komputer.

Bahasa Komputer dapat dibagi atas tiga tingkatan:

1. Leksikal

- Merupakan tingkatan yang paling rendah.
- Yaitu bentuk icon pada layar.
- Pada bahasa manusia, ekuivalen dengan bunyi dan ejaan suatu kata.

2. Sintaksis

- Yaitu urutan dan struktur dari input dan output.
- Pada bahasa manusia, ekuivalen dengan grammar suatu kalimat.

3. Semantik

- Yaitu arti dari percakapan yang berkaitan dengan pengaruhnya pada struktur data internal komputer dan/atau dunia eksternal.
- Pada bahasa manusia, ekuivalen dengan arti yang berasal dari partisipan dalam percakapan.

Dalam user interface, istilah dialog hampir mirip dengan tingkat sintaksis, tetapi juga meliputi sifat-sifat leksikal.

Dialog Manusia – Komputer:

- Berbeda dengan dialog antar manusia pada umumnya, dialog dengan komputer biasanya terstruktur dan terbatas.
- Beberapa ciri-ciri dari dialog terstruktur yang nantinya ditemukan dalam dialog komputer:
 - Menyebutkan beberapa hal tertentu secara berurutan.
 - Beberapa bagian dari dialog dilakukan secara bersamaan.
 - Dialog berikutnya tergantung pada respon dari partisipan.
 - Dialog terstruktur biasanya tidak langsung menuju pada arti kata-katanya / semantik tapi pada level sintaksis.

Proses Perancangan Dialog

Advice:

1. Rangkaian dialog menggambarkan struktur tugas.
 2. Beberapa rangkaian dialog tambahan digunakan untuk user support, misalnya: help system, tutorial sub-sistem.
 3. Rangkaian dialog diurutkan sesuai dengan struktur tugas.
- Prinsip yang digunakan dalam desain dialog adalah membagi sistem menjadi beberapa bagian yang disebut module.
 - Biasanya user access bukan merupakan bagian dari task description, tapi harus dimasukkan dalam sistem yang baru.
 - Empat hal utama dalam desain yang harus diperhatikan dalam GUI metaphor:
 - Pemilihan dan representasi dari conceptual metaphor.
 - Representasi dari obyek interaktif dalam metaphor.
 - Perancangan manipulasi untuk mengimplementasikan user action.
 - Desain micro-metaphor untuk control action dan representasi dari commands.

Dalam mendesain dialog, diperlukan deskripsi yang terpisah dari program secara keseluruhan.

Mengapa perlu digunakan notasi deskripsi dialog yang terpisah?

- Agar mudah di analisa.
- Pemisahan elemen-elemen interface dari logika program (semantik).
- Apabila notasi dialog ditulis sebelum program dibuat, maka notasi tersebut bisa membantu desainer untuk menganalisa struktur yang diajukan.

Notasi Diagramatik

- Notasi Diagramatik paling sering digunakan dalam desain dialog.
- Kelebihan: memungkinkan desainer untuk melihat secara sekilas struktur dialog.
- Kekurangan: sulit untuk menjelaskan struktur dialog yang lebih luas dan kompleks.

Contoh Notasi Diagramatik:

- State Transition Networks (STN)
- Hierarchical STN
- Harel's State Charts
- Traditional Flow Diagrams (Flow Charts)
- JSD Diagrams

State Transition Network (STN)

Komponen STN:

- Lingkaran
Menggambarkan "state" dari sistem.
- Tanda Panah
Yang terdapat antara state disebut juga transisi. Tanda panah ini diberi label yang menjelaskan tentang tindakan user yang menyebabkan transisi dan respon dari sistem.

STN dapat menggambarkan beberapa pilihan dialog:

- Urutan tindakan dan respon dari sistem
- Pilihan bagi user
- Iterasi

Hierarchical State Transition Networks

Struktur Hierarchical STN mirip dengan STN, namun memiliki tambahan berupa gabungan state. Penggunaan hirarki STN ini, seperti menggabungkan beberapa STN ke dalam satu diagram besar, sehingga model ini dapat digabungkan untuk sistem-sistem yang besar.

STN Hirarki dapat digunakan untuk mendeskripsikan suatu sistem yang lengkap.

Harel's State Charts

Dapat digolongkan sebagai kelompok STN. Dibangun untuk menspesifikasikan secara visual sistem reaktif yang kompleks dan mampu mengakomodasi masalah seperti concurrency dan escape.

Flowchart

- Flowchart digunakan untuk mendeskripsikan dialog yang sederhana.
- Flowchart mudah dimengerti dan simple.

Jackson Diagram

JSD (Jackson Structured Design) diagram digunakan untuk berbagai aspek dari analisis tugas dan notasi dialog.

Link to Programming Language

- Sequence dalam notasi dialog dapat langsung ditransformasikan ke bahasa pemrograman dalam bentuk sequence dan selection.
- Modulary dalam desain dialog.

JENIS-JENIS DIALOG

1. Command Language

- Merupakan paradigma user interface yang pertama kali.
- Contoh: MS-DOS shell, UNIX shell, dBase.
- Keuntungan:
 - Lebih cepat.
 - Dapat melakukan kegiatan meskipun di luar batas.
 - Perulangan
 - Implementasi yang mudah dan hemat.
- Tujuan:
 - Konsistensi.
 - Penamaan dan pemberian singkatan yang baik.
 - Melakukan tugas perancangan dapat membantu meminimalkan kekurangan.

2. WIMP (Window, Icon, Menu, Pointer)

Berfokus pada Menus, Buttons, Forms, Icons.

3. Direct Manipulation

Definisi:

- Kesenambungan tampilan dari objek-objek dan aksi-aksi yang dilakukan.
- Cepat, dapat dikembalikan, peningkatan aksi dimana efeknya dapat segera terlihat.
- Penggantian dari sintaks command language dengan manipulasi langsung pada objek.

Keuntungan:

- Mudah dipelajari dan diingat, khususnya untuk pemula.
- WYSIWYG
- Fleksibel
- Menyediakan konteks dan tampilan yang cepat dari feedback, sehingga user dapat mengetahui apakah tujuan telah tercapai.
- Membatasi beberapa jenis kesalahan yang dapat terjadi.

Kekurangan:

- Penggunaan seluruh ruang pada layar.
- Harus mempelajari arti dari komponen-komponen yang tampak pada layar.
- Tampilan visual mungkin dapat disalah artikan.
- Penggunaan mouse dapat lebih lambat daripada pengetikan.
- Tidak ada penjelasan otomatis.
- Tidak bagus pada: pengulangan, penyimpanan history, tugas tertentu (seperti merubah semua huruf cetak miring menjadi cetak tebal), variabel.

4. PDA dan Pen

- Menjadi lebih umum dan penggunaan yang lebih luas.
- Tampilan yang lebih kecil (160x160 atau 320x240).
- Tombol-tombol yang lebih sedikit, interaksi melalui "pen" (atau stylus).
- Peningkatan → wireless, warna, memory yang lebih besar, CPU dan OS yang lebih baik.
- Palmtop vs Handheld.

5. Speech dan Natural Language

Speech (Suara)

Adalah getaran pada suara menimbulkan bunyi “ahh”.
Mulut, tenggorokan, bibir membentuk bunyi.

Input speech:

- Pengenalan speaker.
- Pengenalan suara.
- Pemahaman bahasa natural.

Natural Language:

- Memberi arti pada kata-kata.
- Input dapat berupa suara atau dari keyboard.

Keuntungan:

- Mudah dipelajari dan diingat.
- Lebih kuat.
- Cepat, efisien (tidak selalu).
- Layar yang kecil.

Kekurangan:

- Belum dapat bekerja dengan baik.
- Mengasumsikan pengetahuan dari permasalahan.
- Membutuhkan keahlian pengetikan. Peningkatan tidak terlihat.
- Implementasi membutuhkan biaya yang mahal.

USER INTERFACE SOFTWARE

Alat Bantu User Interface:

- Aplikasi apa yang biasa digunakan oleh programmer untuk membuat program.
- Kombinasi dari objek-objek interface dan perilaku manajemen.
- Sekarang sudah object-oriented.
- Pustaka dari komponen-komponen perangkat lunak dan routines yang digunakan oleh programmer:
 - Xwindows : Xtoolkit & Motif
 - Macintosh : Mac Toolbox, MacApp
 - Windows : Windows Developer's Toolkit
 - Java : Swing

Bagaimana cara kerja toolkit?

- Apa yang sebenarnya disediakan olehnya?
- Bagaimana mengaturnya?
 - User mengadakan aksi, berinteraksi dengan interface.
 - Aksi-aksi tersebut harus disampaikan kepada aplikasi dalam cara-cara yang penuh arti.
 - Aplikasi melakukan aksi yang sesuai, mungkin memperbaharui tampilan.

Model Seeheim

Model Percakapan

Model Objek

- UI adalah kumpulan objek yang saling berinteraksi.
- User secara langsung memanipulasi objek-objek tersebut.
- Objek-objek bertanggung jawab untuk mentransmisikan aksi-aksi user ke aplikasi dengan cara yang berguna.

OBJECT-ORIENTED SYSTEMS

- Pemrograman GUI Java dikerjakan dengan AWT dan Swing.
- Lebih banyak model terdistribusi.
- Aksi utama disini adalah mengirimkan event ke objek-objek sebagai pesan.
- Pendelegasian yang penting → dapat membuat beberapa objek tertentu bertanggung jawab untuk menangani event.

GUI Builder Tools (Alat Pembuat GUI)

- Mengapa membangun interface grafis dengan perintah-perintah tekstual?
- Mengapa tidak menunjukkan apa yang Anda sukai?
- Visual builder tools: Visual Basic, Visual C++, Borland Delphi, Symantec Café.

Metode-metode Tool

- Area kerja (interface yang sedang dibangun).
- Pindahkan dan letakkan interaktor ke dalam area kerja.
- Spesifikasi lokasi, warna, tampilan, dsb.
- Biasanya menyediakan mode Build/Pengujian.