

DISEÑO

El diseño, una de las actividades del Ciclo de Vida del Proyecto Estructurado, es un proceso iterativo que toma un modelo lógico de un sistema, junto con un conjunto de objetivos fuertemente establecidos para este sistema, y produce las especificaciones de un sistema físico que puedan satisfacer esos objetivos.

El diseño de datos, el diseño arquitectónico, el diseño procedimental y el diseño de la interfaz, que establece la disposición y los mecanismos para la interacción hombre máquina y, que según Yourdon comienza a definirse en el Modelo de Implantación del Usuario, son parte de la actividad de diseño.

El diseño involucra diferentes tareas que permiten el desarrollo de un modelo físico, denominado Modelo de Diseño.

Son tareas del diseñador:

- Sincronizar el modelo esencial con una configuración de procesadores.
- En cada procesador, asignar procesos y datos a diferentes tareas.
- En cada tarea, organizar procesos en una jerarquía de módulos.

El **Modelo de Diseño** está compuesto por:

- Modelo de implantación de sistemas:
 - *Modelo del procesador:* Asignación de procesos y almacenes a recursos apropiados de hardware (procesadores y unidades de almacenamiento). Para la asignación el diseñador considera diversos factores, tales como costo, eficiencia, seguridad, confiabilidad y restricciones políticas y ambientales.
 - *Modelo de tareas:* Asignación de los procesos y almacenes a las tareas individuales de cada procesador.
- Modelo de implantación de programas: Representación mediante *diagramas de estructura* de la organización jerárquica de módulos dentro de una tarea individual.

Objetivos del diseño:

Rendimiento: cuán rápido permitirá el diseño realizar el trabajo del usuario, dado un recurso de hardware.

Control: la medida en que el diseño está protegido contra errores humanos, máquinas defectuosas o daños intencionales.

Cambiabilidad: la facilidad con la cual el diseño permite modificar el sistema.

Por lo general, acontece que estos tres factores trabajan unos en contra de los otros.

Consideraciones de rendimiento.

El rendimiento normalmente se expresa en términos de:

- Volumen de procesamiento (transacciones l cálculo por hora)
- Tiempo de corrida (para una tarea en lote, donde se procesa la misma cantidad de trabajo en cada corrida)
- Tiempo de respuesta (el tiempo entre que se pulsa la tecla de "entrada" en una terminal y el comienzo de la aparición de la respuesta del computador en dicha terminal)

Consideraciones sobre el control.

Dependiendo de la naturaleza del sistema y la cantidad de dinero en juego, el diseñador necesitará introducir diversos tipos de controles:

- El uso del dígito verificador sobre números predeterminados. Muchos bancos asignan el último dígito del número de la cuenta personal como dígito verificador. Cada vez que se procese en el futuro el n° de cuenta los dígitos del no, tal como se ingresan, se utilizan para recalcular el dígito verificador, y la respuesta se compara con el dígito verificador ingresado. Si no concuerdan, uno o más dígitos han sido ingresados incorrectamente.
- El uso de totales de control de lote. Como parte de la validación de un lote el diseñador deberá especificar que el computador calcule el total del lote y lo compare con el total calculado manualmente.
- La creación de libros diarios y líneas de auditoría. El diseñador puede especificar que cada acción que entre al sistema se grabe en un registro o archivo diario que podrá ser leído por los auditores. Algunas veces se llevará un diario de archivo de acciones el que registrará cada cambio de cualquier archivo. Copias de respaldo de los archivos o de puntos de control también representan un uso para seguridad y control.
- La limitación de los accesos a los archivos. ("¿Quién puede acceder a estos datos?", "¿Quién está autorizado a modificar estos datos?"). Los accesos pueden limitarse mediante el empleo de contraseñas, o palabras clave, las que deberán ingresarse antes de que una persona pueda utilizar una terminal.

Consideraciones sobre cambiabilidad.

La cambiabilidad de un sistema es muy importante, si se pudiera encontrar una manera de diseñar el sistema más modificable, sería fácil de corregir, fácil de adaptar a software y hardware cambiantes, y fácil para incorporarle los cambios y mejoras solicitadas por los usuarios.

- Los sistemas no son estáticos.
- Detectar mantenimiento (cambios en el mundo externo, cambios para ajustar a nuevo hardware/software, corrección de errores en producción) y perfeccionamiento (cambios que proveen funciones adicionales o mejoras) de un sistema.
- Considerar la cambiabilidad como la medida de tiempo que lleva hacer un cambio en el sistema (ya sea suprimir un error o instalar una mejora) en forma oportuna.

Diseño Estructurado para Cambiabilidad.

Los sistemas más fáciles de cambiar son aquellos que están constituidos por módulos manejablemente pequeños, cada uno de los cuales es independiente (en la medida de lo posible) de manera que puedan sacarse del sistema, cambiarse y reponerse sin afectar al resto del sistema.

La tarea del diseñador es formar los módulos y diseñar sus interconexiones para minimizar la posibilidad del efecto de onda (el efecto del cambio original provoca "ondas" a lo largo del sistema). Algunos sistemas están tan interconectados que se vuelven imposibles de mantener.

La cambiabilidad es el grado con el cual el diseñador logra aislar la función en la menor cantidad de módulos posibles.

Las funciones deben estar contenidas dentro de cajas negras, donde la función producirá siempre resultados predecibles a partir de un conjunto de datos que pasen a través de la misma.

Acoplamiento de los módulos.

Uno de los objetivos del diseño, en cuanto hace a la cambiabilidad, es tener el menor acoplamiento posible de módulos, sin afectar el funcionamiento del sistema.

- Acoplamiento de datos. Un módulo le transfiere datos a otro como parte de la invocación o del retorno del control. Este es el mejor acoplamiento debido a que es mínimo.
- Acoplamiento de control. Cada vez que un módulo de trabajo lee, graba, o entra de alguna manera en contacto con el mundo exterior, tiene que informar a su jefe lo ocurrido, puede detectar una transacción inválida, etc. Esta información de retorno involucra transferir una variable de control.

Cohesión.

Al mismo tiempo que se minimiza el acoplamiento debemos especificar módulos elegidos y formados lo mejor posible.

Un módulo altamente cohesivo, cuyas partes contribuyan todas a una sola función, probablemente no necesite mucho acoplamiento con otros módulos; un módulo pobremente cohesivo, a menudo una combinación de partes no relacionadas, probablemente necesite un alto acoplamiento con otros. El bajo acoplamiento y la alta cohesión van unidos y viceversa. Podemos identificar seis tipos de cohesión (desde el peor al mejor):

- Cohesión coincidente (no puede apreciarse que los elementos del módulo lleven a cabo ninguna función definible)

- Cohesión lógica (varias funciones semejantes, pero ligeramente diferentes, están combinadas juntas, configurando un módulo más compacto que si cada función fuera programada separadamente)
- Cohesión temporal (son módulos con una cantidad de funciones cuyo único elemento común es el de ser ejecutados al mismo tiempo)
- Cohesión de procedimiento (un módulo para cada proceso del DFD. Dentro de cada módulo se ejecutan varias funciones, relacionadas a través del flujo de control entre ellas)
- Cohesión de comunicación (las funciones operan todas sobre la misma corriente de datos además de ser cohesivas de procedimiento. Ej.: "Calcular solución e imprimir resultado").
- Cohesión funcional (realiza una y solo una función identificable)

Bibliografía

Gane, Chris; Sarson Trish. Analisis Estructurado de Sistemas. El Ateneo. Buenos Aires, 1987.

Yourdon, Edward. Análisis Estructurado Moderno. Prentice-Hall Hispanoamericana. México 1993.