

Diseño estructurado

Es diseño estructurado es una técnica de diseño que permite el paso entre el análisis del problema y la instrumentación del mismo. Consiste en un conjunto de reglas que aplicadas a una especificación, permite derivar una arquitectura del software.

Características

- Permite obtener una solución a partir de la definición del problema, ya que la solución tiene los mismos componentes e interrelaciones que en el problema original.
- Simplifica el problema, lo subdivide y cre una división del problema en cajas negras, organizadas en jerarquías.
- Ofrece un conjunto de estrategias para desarrollar una solución de diseño a partir de análisis. Permite, mediante una serie de estrategias, derivar una arquitectura del software a partir del DFD que define la solución propuesta al problema.
- Utiliza técnicas gráficas que facilitan la especificación del diseño y la comunicación de éste. La diagramación es imprescindible en cualquier ingeniería. En el diseño estructurado, se utiliza el **diagrama de estructuras**, también pueden utilizarse las **tablas de interfaz**.
- Criterios para evaluar la calidad de la solución respecto al problema.

Diagrama de Estructuras (DEC)

Características

Puede observarse directamente la jerarquía de control, la comunicación entre módulos, los datos de entrada/salida, los datos de control de entrada/salida. Para que un diagrama de estructuras sea fácil de entender, los nombres de los módulos, datos y flags deben ser lo más significativos posibles. Los flags no acostumbran a representarse, excepto en el análisis de transacción. Una diapositiva del diagrama de estructuras, al igual que en cualquier proceso de ingeniería, es la posibilidad de abstracción. Cuando la comunicación entre módulos es muy compleja, se pueden abstraer en el diagrama los componentes de la comunicación. No obstante, hay que tener en cuenta que se debe conseguir la máxima cohesión y el mínimo acoplamiento.

Módulos

AECC: Parte lógica separable de un programa. Yourdon: Es una secuencia contigua de sentencias de programa, limitada por delimitadores y que tiene un identificador global. Fenton: Cualquier objeto que, en un nivel de abstracción dado, queramos considerar como un concepto simple. D.E: Es aquella parte del código que se puede llamar.

Visiones del módulo

Externa

- Entrada
- Salida
- Función Interna

- Mecanismos
- Datos internos

Conexión y comunicación

Conexión: líneas que conectan el módulo llamador con el llamado. Comunicación: Datos de procesamiento, la relación de ellos con el problema, datos importantes del mundo exterior
Flags: valores de condición, importantes internamente.

Tabla de interfaz

Especifica los parámetros, datos o flags, que se pasan entre módulos. La tabla de interfaz especifica, por lo tanto, la interfaz de los módulos.

Componentes

- Módulo llamado
- Parámetros formales
- Parámetros de entrada
- Parámetros de salida
- Uso del parámetro
- Significado del parámetro

Letra	Significado				
P	El parámetro es procesado				
M	El parámetro es modificado				
T	El parámetro es transferido a un módulo subordinado				
C	El parámetro es usado como una variable de control				
I	El parámetro es transferido a un módulo subordinado y modificado en éste				
Módulo	Parámetro Formal	Entrada	Salida	Uso	Significado
Factorial	Número	S	N	P	Factorial a calcular
Factorial	Resultado	N	S	M	Factorial calculado

Uso

Las tablas de interfaz no son independientes de los diagramas de estructura, sino que los complementa. Permiten al diseñador mejorar la claridad de los diagramas de estructuras mediante una especificación más rigurosa de las interfaces. Permite desglosar una comunicación de datos definida en un diagrama de estructuras.

Estrategias de Diseño

Permite una transformación del DFD incluido en la especificación de requisitos a una descripción de diseño de la arquitectura del software. Dos técnicas distintas: Análisis de transformación y análisis de transacción

Análisis de transformación

La información llega al sistema mediante caminos que transforman los datos externos (flujos de llegada). En el centro del sistema, en el núcleo, ocurre una transformación. Los datos de llegada transformados circulan por caminos que conducen a la salida.

Análisis de transacción

La información llega al sistema mediante caminos que transforman los datos externos (flujos de llegada). En el centro del sistema, el núcleo, ocurre un encaminamiento del flujo de entrada. Los datos de llegada encaminados circulan alguno de los distintos caminos de acción (flujo de salida).

Técnica

En un DFD real, los procesos formarán combinaciones de transformación-transacción. Deberán realizarse ambos análisis. Las zonas de transformación y transacción no están definidas por la forma. Se debe analizar el flujo de datos en el DFD para saber ante qué estamos.

Análisis de transformación

1. Revisar el modelo del sistema

Se debe obtener el DFD de la especificación del sistema. Si no existiese, debería crearse a partir de las especificaciones funcionales del sistema. El DFD debe tener el suficiente nivel de detalle, típicamente una profundidad de tres o más niveles. Es necesario para poder aplicar el análisis de transformación con el detalle necesario.

2. Determinar las características del DFD

En general, cualquier DFD puede realizarse únicamente con caminos de transformación. Sin embargo, para la aplicación de esta técnica conviene escoger aquellos DFD con características claras de transformación. Si existe algún proceso con salidas exclusivas, probablemente estamos ante un DFD de transacción.

3. Aislar el centro de transformación

El centro de transformación contiene las funciones esenciales del DFD. Los límites de flujo de llegada y del flujo de salida están abiertos a interpretación, se pueden derivar distintas alternativas de diseño.

4. Realizar un "Primer Nivel de Factorización"

A todos los módulos se les debe asignar un nombre significativo que defina lo que van a realizar sus módulos subordinados.

5. Realizar un "Segundo Nivel de Factorización"

Se progresa desde el centro de transformación en dirección contraria al flujo de llegada, colocando los procesos del DFD como módulos subordinados. Con el flujo de salida debe realizarse el mismo proceso. Al final de la jerarquía, se introducen módulos predefinidos que proporcionen las entradas y salidas necesarias.

6. Refinar la arquitectura del sistema

Aumentar o disminuir los módulos. Se deben especificar los datos comunicados entre módulos de la arquitectura del sistema. Las flags no acostumbran a representarse, aunque en algún caso, por claridad, pueden ser necesarios. Estos datos se derivan por refinamiento de los flujos del DFD.

7. Revisar el diseño

Se debe revisar el diagrama de estructura para comprobar que el diseño derivado es correcto. Por ejemplo, se puede realizar una simulación de la ejecución de la jerarquía obtenida, para comprobar si es consistente con el DFD. De este forma, podrá comprobarse que el orden de ejecución es correcto.

Análisis de transacción

1. Revisar el modelo del sistema

2. Determinar las características del DFD

3. Identificar el centro de transacción

Se debe determinar que proceso constituye el centro de transacción. Este proceso es, normalmente, identificable a simple vista, ya que de él emanan varios caminos independientes de tratamiento. Por último, debe aislarse el flujo de llegada y evaluar cada camino de acción. Los caminos de acción pueden pertenecer a un camino tipo transformación o a un camino tipo transacción. Cada camino debe identificarse individualmente para realizar el segundo nivel de factorización.

4. Realizar un "Primer Nivel de Factorización"

5. Realizar un "Segundo Nivel de Factorización"

Se desarrolla el camino de entrada al igual que en el análisis de transformación. Cada camino de acción se desarrolla según su tipo. Transformación o Transacción.