Diseño de Software

- **El Diseño no puede ser definido solo puede explicarse en base a los distintos puntos de vista y tareas que realizan los diseñadores del software**
- **Basado en la traducción de Sommerville 5ta ed. Acotaciones teóricas del Lic. Domingo F. Donadello Universidad Nacional de La Matanza**
- **Ciclo lectivo 2004**

Diseño de Software

- Sin embargo, podemos decir que el Diseño es la Interfase entre las especificaciones de requerimientos y la construcción de soluciones de software que satisfagan dichos requerimientos del software

Objetivos

- Introducir el proceso de diseño de software
- Describir las diferentes fases dentro del proceso del diseño
- Mostrar las distintas aproximaciones de diseño, funcional y orientada a objetos y como las citadas estrategias de diseño orientadas a objetos y funcional son complementarias
- Discutir algunos de los atributos necesarios de calidad del diseño

Tópicos a ser desarrollados

- El proceso de diseño y los métodos, técnicas y herramientas de diseño de software
- Estrategias de diseño que incluyen el diseño orientado a objetos y la descomposición funcional
- Atributos de calidad del diseño

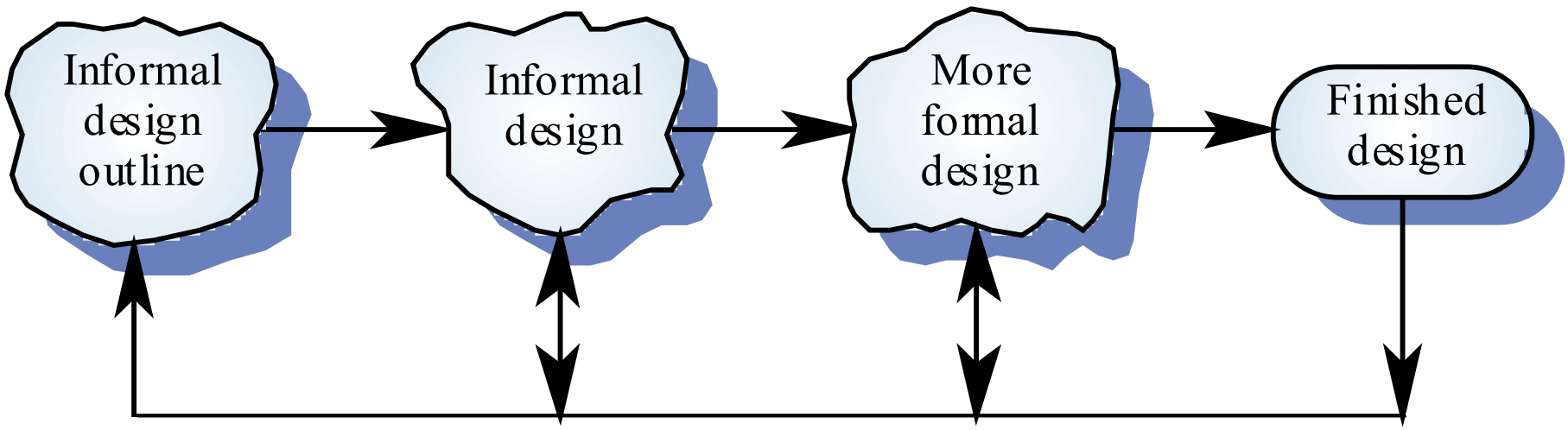
Etapas del diseño

- Entendimiento del problema
  - Visualizar el problema desde varios ángulos y descubrir los requerimientos del diseño
- Identificar una o mas alternativas de solución
  - Evaluar posibles soluciones y escoger las mas apropiadas de acuerdo a la experiencia del diseñador y los recursos disponibles
- Describir abstracciones de la solución
  - Utilizando notaciones descriptivas gráficas, formales o otras que permitan describir los componentes del diseño
- Repetir el proceso para cada abstracción identificada hasta que el diseño este expresado en términos sencillos y pueda ser input a la construcción del software

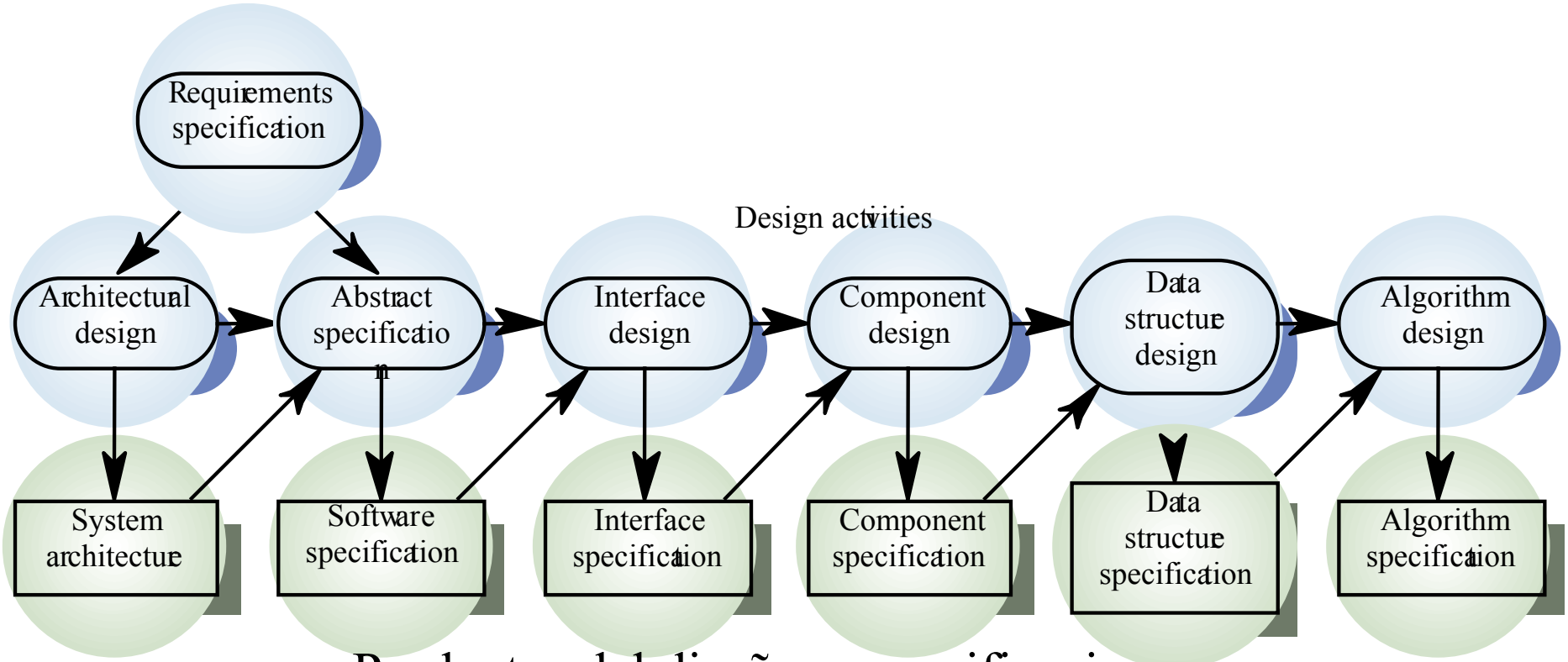
El proceso de diseño

- Cualquier diseño debe ser modelado como una gráfica dirigida hecha de entidades con atributos los cuales participan en relaciones
- El sistema debe estar descrito a distintos niveles de abstracción
- El diseño ocurre en etapas que se traslapan y no necesariamente son etapas secuenciales.

Del diseño informal al diseño formal



Fases en el proceso de diseño

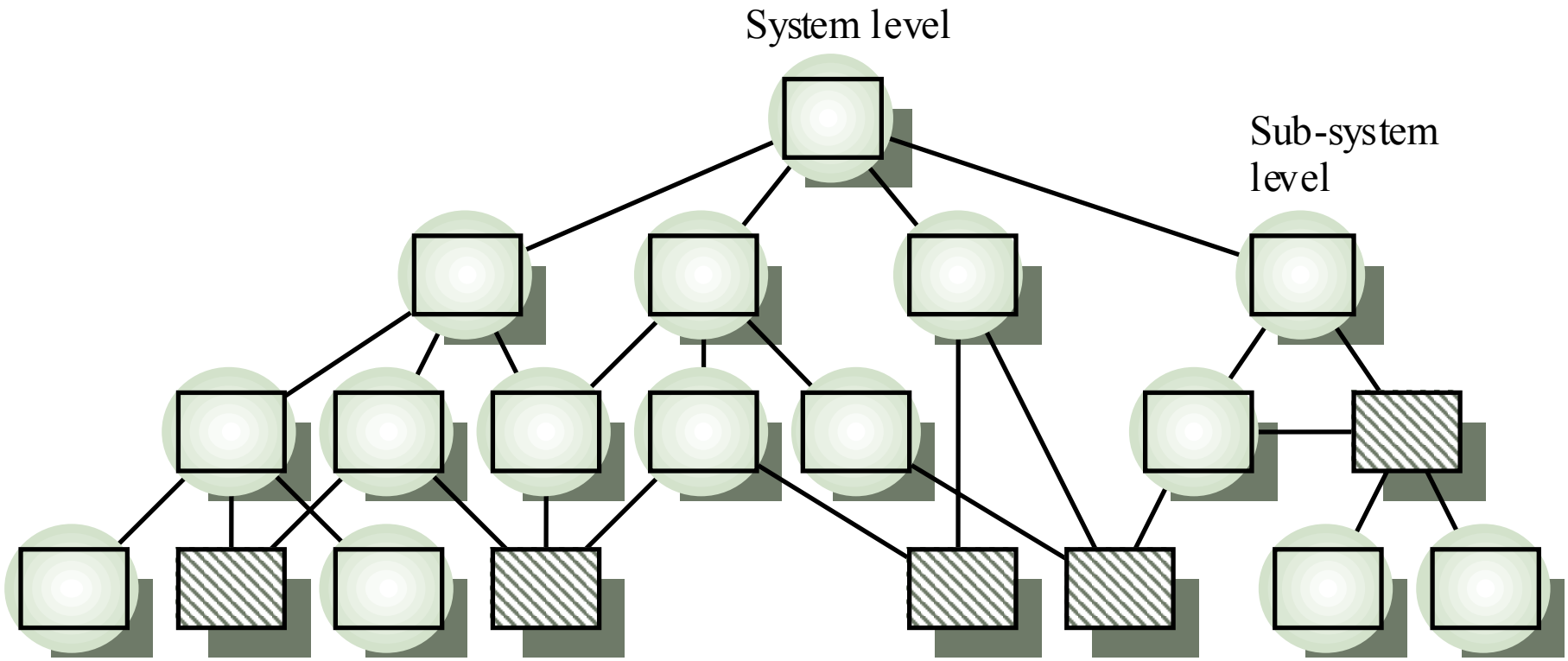


Productos del diseño - especificaciones

Fases del Diseño

- *Diseño de la Arquitectura.* Identificar subsistemas y alocarlos en componentes de hardware
- *Especificación de abstracciones* Especificar subsistemas
- *Diseño de Interfaces* Describir las interfaces de los subsistemas con el usuario y entres subsistemas
- *Diseño de Componentes* Descomposición de subsistema en componentes
- *Diseño de las estructuras de datos* Diseñar las estructuras de datos internas y externas (base de datos)
- *Diseño de los algoritmos* Diseñar algoritmos para las funciones de los problemas a resolver

Estructura de diseño jerárquica



Diseño Top-down

- Necesidad de particionar la complejidad de los sistemas, una manera es ir de lo general a lo particular aplicando la aproximación top-down
- En principio el diseño top-down involucra comenzar con los componentes mas altos en la jerarquía y desarrollar el trabajo hacia abajo en los subsecuentes niveles
- En la practica, los sistemas grandes nunca se construyen con esta técnica. Alguna partes se diseñan antes de otras. Los diseñadores reutilizan experiencia (y en algunos casos componentes) durante el proceso de diseño

Métodos de Diseño

- Los métodos estructurados son conjuntos de notaciones que permiten expresar el diseño del software en términos de funciones que transforman datos y proveen guías para la creación y especificación del diseño
- Existen métodos ampliamente conocidos como el diseño estructurado (Meyers, Constantine y Yourdon) y JSD (Metodo de Jackson)
- Pueden aplicarse exitosamente debido a que soportan notaciones estándares que permite que el diseño siga una forma estándar relacionada con el problema.
- Los métodos estructurados están soportados en la mayoría de herramientas CASE
- Los elementos básicos son DFD, DER, Carta Estructurada de módulos, DTE

Componentes de los Métodos

- Muchos métodos soportan varias vistas del sistema
- El flujo de datos (diagramas de flujo de datos) muestran las transformaciones de los datos
- Las entidades-relación describen las estructuras de datos lógicas
- Las vistas estructurales muestran los componentes del sistema y sus interacciones
- Los Diagramas de Estado muestran el comportamiento de los componentes en tiempo de ejecución

Deficiencias de los Métodos

- En la práctica, lo que existen son mas bien guías y pautas en lugar de métodos en el estricto sentido matemático ya que distintos diseñadores crean distintos diseños del mismo sistema y todos los diseños llevan a soluciones que funcionan
- En general los métodos no ayudan mucho en la fase inicial del diseño. En vez de esto, ayudan al diseñador a estructurar y documentar sus ideas de diseño
- Luego el diseño del software aún hoy sigue siendo una actividad intelectual donde priva la creatividad y capacidad de abstracción del diseñador del software

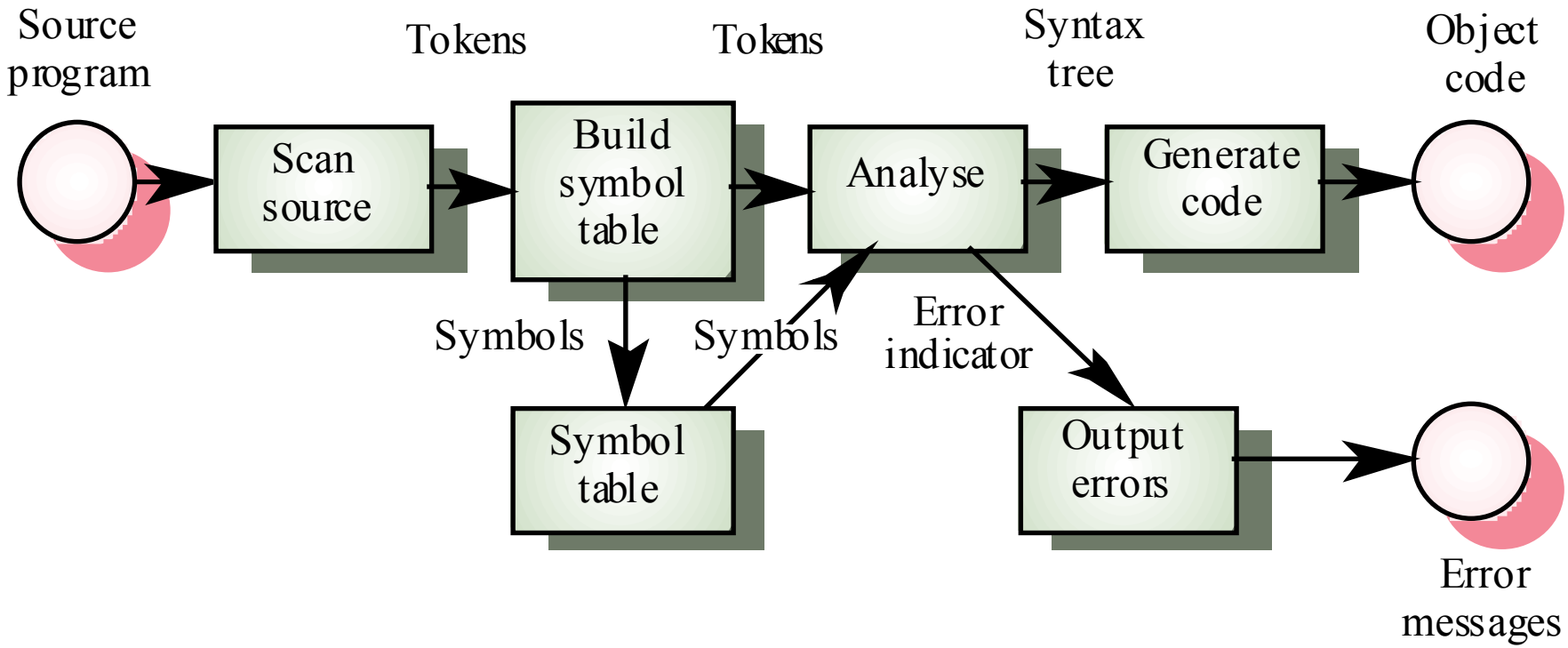
Descripción del Diseño

- *Notaciones gráficas*. Utilizadas para desplegar las relaciones entre los componentes
- *Lenguajes de descripción de programas*. Basadas en lenguajes de programación pero con mas flexibilidad para representar conceptos abstractos
- *Texto informal*. Descripción en lenguaje natural
- Todas estas notaciones pueden usarse para el diseño de sistemas grandes

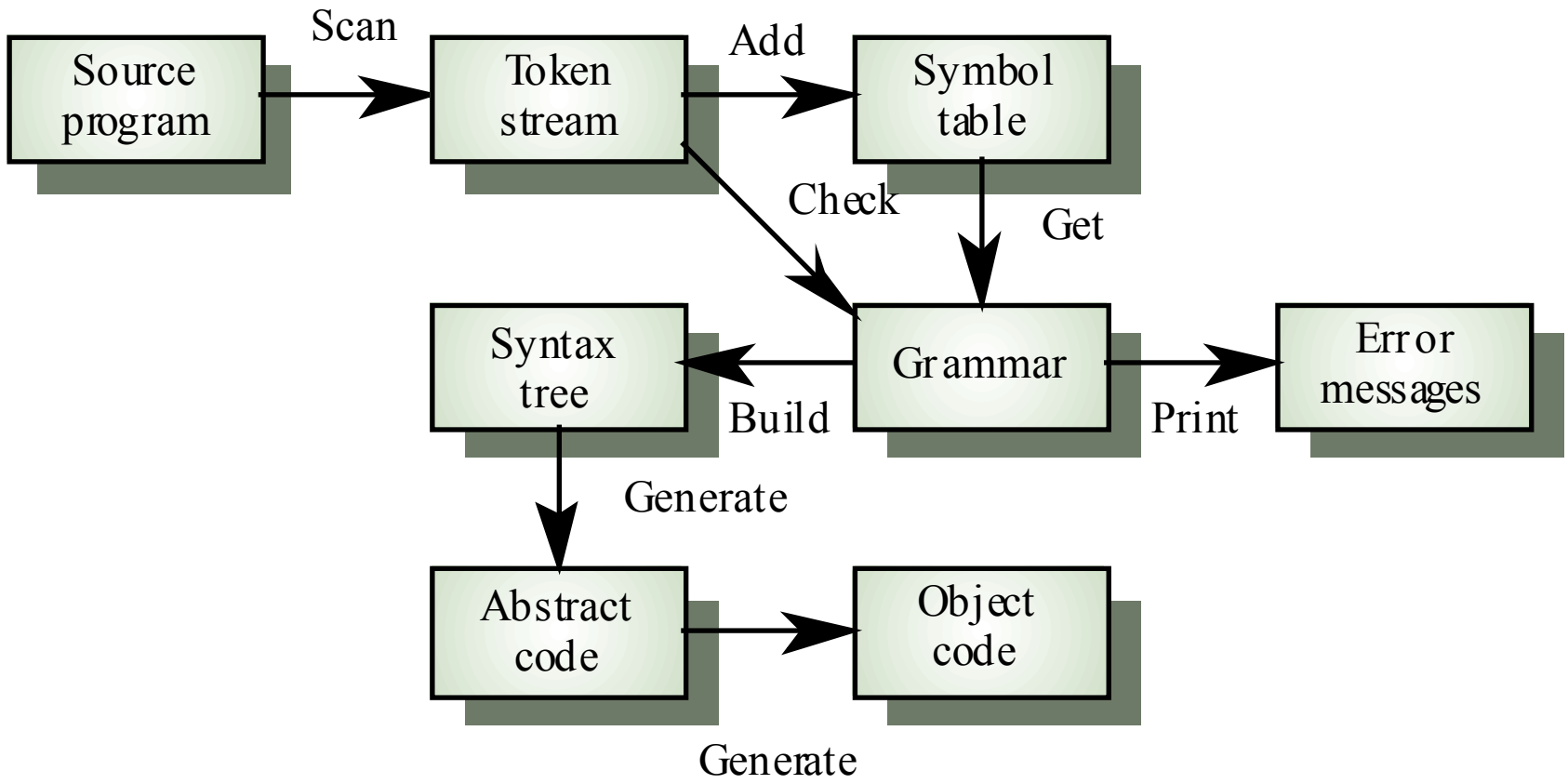
Estrategias de Diseño

- Diseño Funcional
  - El sistema es diseñando desde un punto de vista funcional. El estado del sistema es centralizado y compartido entre las funciones que operan en ese estado
  - Diseño orientado a Objetos
  - El sistema es visualizado como una colección de objetos que interactúan y colaboran entre si. EL estado del sistema no esta centralizado y cada objeto maneja su propio estado. Los objetos pueden ser instancias de una clase objeto y se comunica mediante métodos de intercambio

Vista Funcional de un compilador



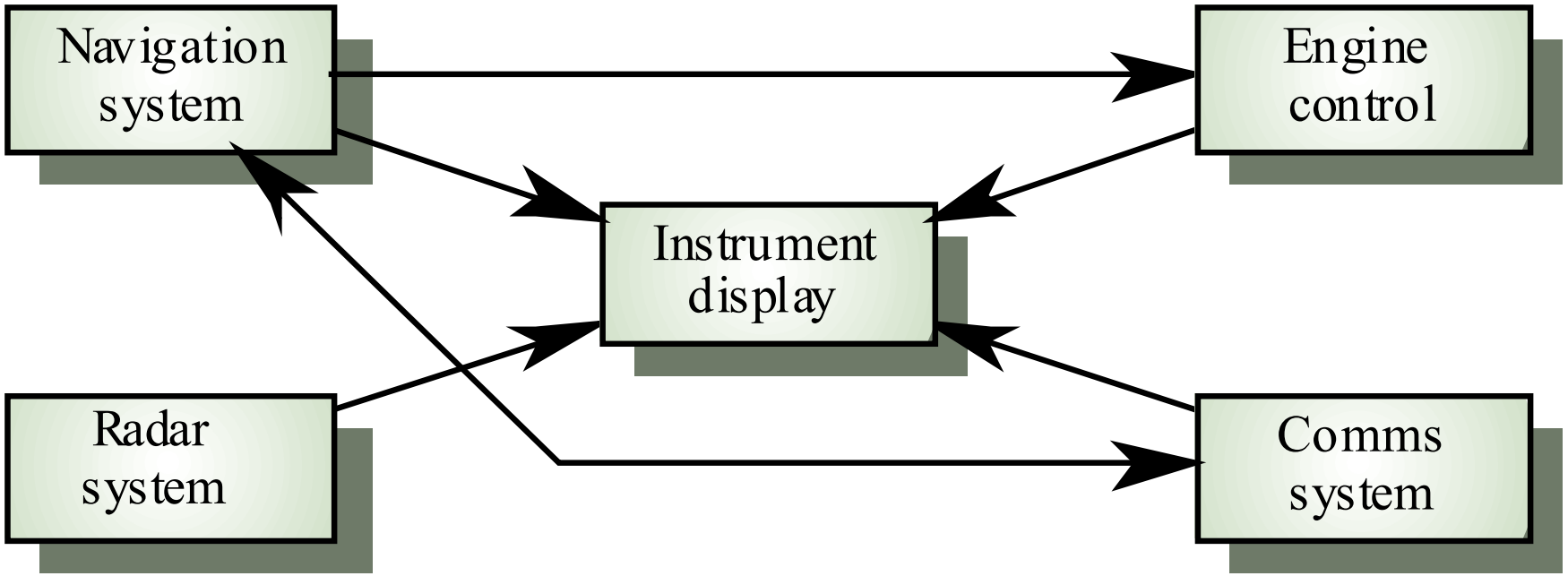
Vista orientada a objetos de un compilador



Estrategia de diseño mixta

- Aunque a veces se sugiere la súperioridad de algún método de diseño, en la practica, los enfoques de diseño orientado a objetos y funcional son complementarios sobre todo en grandes sistemas a desarrollar
- Los buenos Ingenieros de Software deben seleccionar el método mas apropiado para cualquier subsistema que esta siendo diseñando

subsistemas de una aeronave



Objetos de alto nivel

- Sistema de navegación
- Sistema de radar
- Sistema de comunicaciones
- Sistema de instrumentos de despliegue de información
- Sistema de control de motores

Funciones del sistema (a nivel de subsistema)

- Despliega el rastreo que realiza el radar (subsistema de radar)
- Compensa la velocidad del viento (subsistema de navegación)
- Reduce potencia (subsistema de motores)
- Indica emergencias (subsistema de instrumentos)
- Busca frecuencias (subsistema de comunicaciones)

Objetos de bajo nivel

- Estatus de los motores
- Posición de la aeronave
- Medición del Altímetro
- La respuesta del radio
- ...

Calidad del diseño

- La calidad del diseño puede ser un concepto vago. La calidad depende de las prioridades de la organización
- Un buen diseño puede ser el mas eficiente, el mas barato, el mas mantenible, el mas confiable, etc.
- Los atributos discutidos aquí conciernen con la mantenibilidad del diseño
- Las características de calidad son igualmente aplicables a diseño orientados a funciones como a diseños orientados a objetos

Cohesión

- Es una medida que indica que tan bien un componente encaja junto a los demás
- Un componente debe implementar una única entidad lógica o una función
- La cohesión es un atributo deseable de los componentes sobre todo cuando se realizan cambios.
- Pueden identificarse varios niveles de cohesión

Niveles de Cohesión

- Cohesión por coincidencias (débil)
  - Partes de un componente son reunidas
- Asociación lógica (débil)
  - Los componentes que realizan funciones similares son agrupados
- Cohesión temporal (débil)
  - Los componentes que son activados al mismo tiempo son agrupados
- Cohesión procedural (débil)
  - Los elementos de un componente realizan una secuencia de control única

Niveles de Cohesión

- Cohesión de Comunicación (medio)
  - Todos los elementos de un componente operan sobre la misma entrada o produce la misma salida
- Cohesión secuencial (medio)
  - La salida de una parte de un componente es la entrada de otra parte
- Cohesión Funcional (fuerte)
  - Cada parte de un componente es necesaria para la ejecución de una función única
- Cohesión de objeto (fuerte)
  - Cada operación provee una funcionalidad que permite a los atributos de un objeto ser modificados o inspeccionados

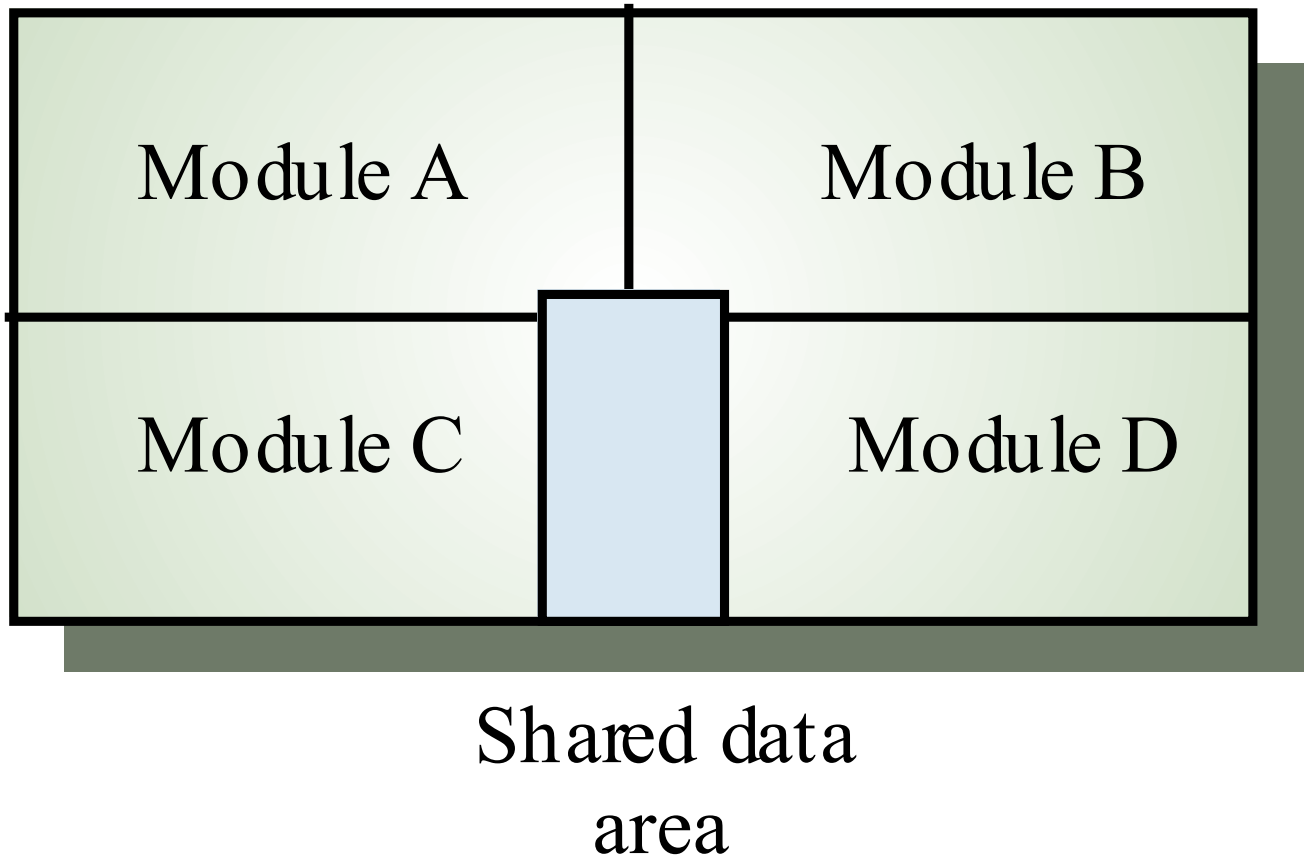
Cohesión visto como un atributo del diseño

- No esta bien definido. A menudo es difícil de clasificar la cohesión
- La herencia de atributos de súper-clases debilita la cohesión
- Para entender un componente, las súperclases y las clases componentes debe ser examinadas

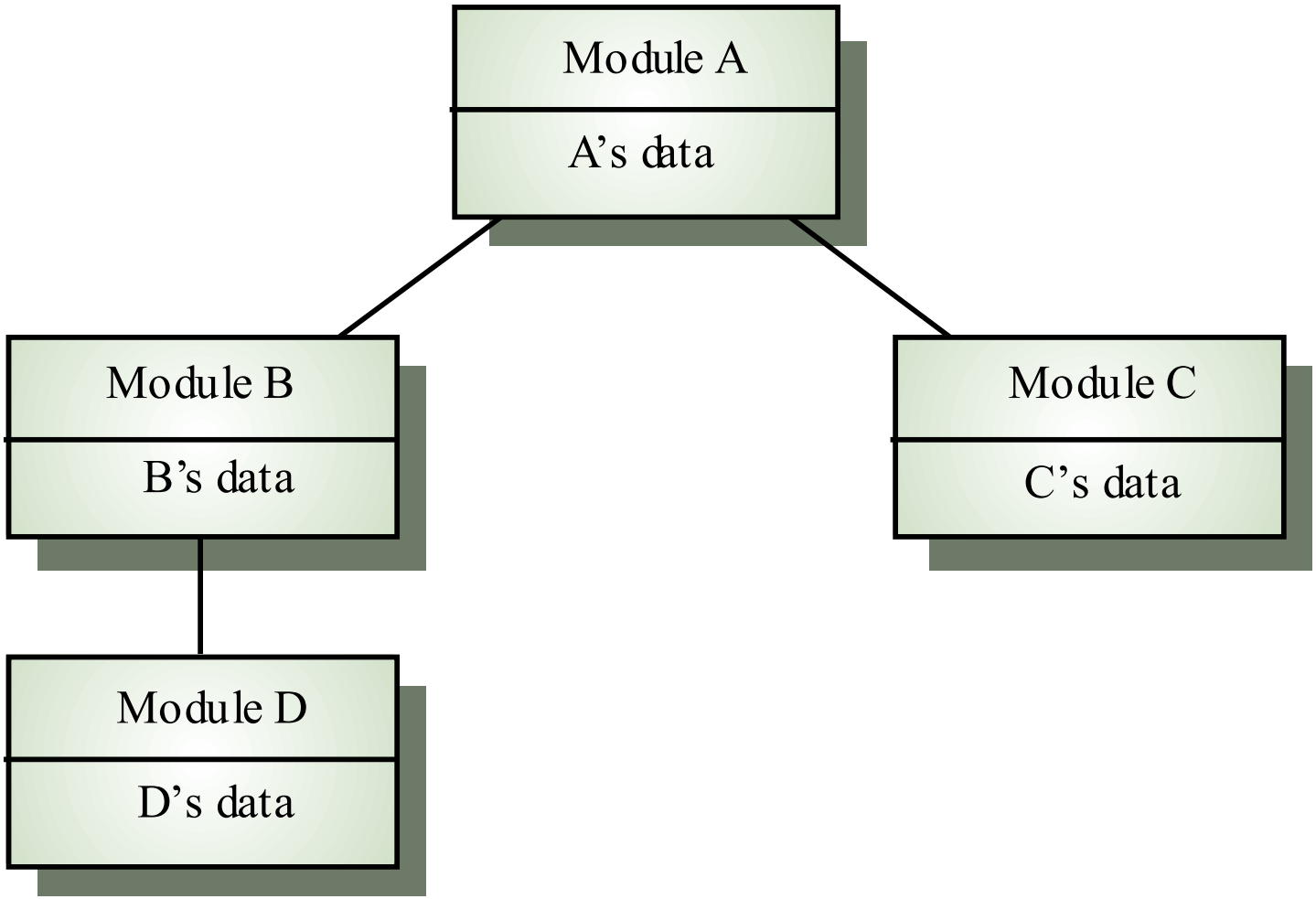
Acoplamiento (Coupling)

- Es una medida de la fuerza de las interconexiones entre los componentes del sistema
- Acoplamiento débil significa que los cambios en los componentes no afectaran a otros componentes
- Las variables compartidas o el intercambio de información llevan al acoplamiento fuerte
- El acoplamiento débil puede lograrse mediante la descentralización de estados (como en los objetos) o mediante el paso de mensajes

Acoplamiento Fuerte



Acoplamiento Débil



Acoplamiento y herencia

- Los sistema orientados a objetos son débilmente acoplados por que no hay estados compartidos y los objetos se comunican usando paso de mensajes
- Una clase de objetos esta acoplada a su súper clase. Los cambios hechos a los atributos o operaciones de una súper clase propagan a todas las subclases. Tales cambios deben de ser cuidadosamente controlados

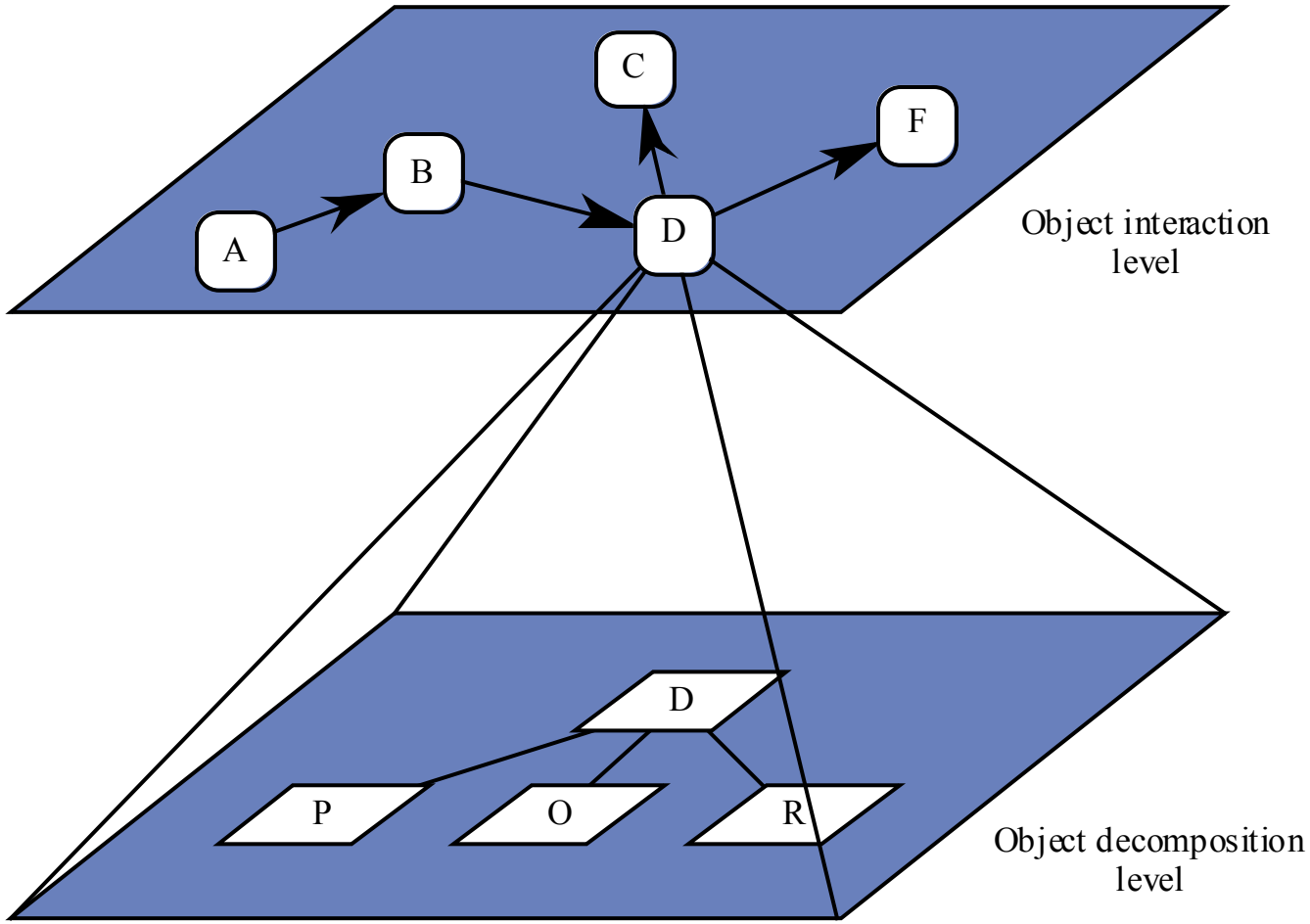
Entendibilidad (Understandability)

- Relacionado con varias características de los componentes
  - *Cohesión*. Pueden ser los componentes comprendidos por si mismos?
  - *Nombrado.* Los nombres usados tienen significado?
  - *Documentación*. El diseño esta bien documentado?
  - *Complejidad*. Se usan algoritmos muy complejos?
- Informalmente, una alta complejidad significa que existen muchas relaciones entre varias partes del diseño, por lo cual es difícil de entender.
- La mayoría de las métricas de calidad del diseño están orientadas hacia la medida en la complejidad. Además de que son de uso limitado

Adaptabilidad

- Un diseño es adaptable si
  - sus componentes están débilmente acoplados
  - esta bien documentado y la documentación esta actualizada
  - existe una correspondencia obvia entre los niveles del diseño (visibilidad del diseño)
  - Cada componente es una entidad auto contenida (fuertemente acoplada)
- Para adaptar un diseño, debe ser posible trazar las ligas de los componentes del diseño de manera que las consecuencias en los cambios puedan ser analizadas

Rastreo del Diseño (traceability)



Adaptabilidad y herencia

- La herencia mejora dramáticamente la adaptabilidad. Los componentes pueden adaptarse sin cambios derivando subclases y modificando las clases derivadas
- A medida que la se incrementa la profundidad en la jerarquía de la herencia, esta se vuelve muy compleja. Por lo cual debe ser periódicamente revisada y reestructurada

CONSIDERACIONES SOBRE EL DISEÑO Y EL DISEÑADOR:

- EL GENIO NO DISEÑA, expresa espontáneamente sus sensaciones
- El GENIO no es CONCIENTE de las fuerzas creativas interiores que posee
- EL MUNDO posee un número limitado de Genios
- El HOMBRE COMUN necesita guías conceptuales y de procedimiento para generar diseños de software

- El DISEÑO NO ES UNA ACTIVIDAD FORMULABLE
- EL DISEÑO NO ES UN PROCESO DETERMINISTICO
- EL DISEÑO NO TIENE LIMITES, SIEMPRE SE PUEDE AVANZAR A UNA VERSION MAS ACPETABLE
- EL DISEÑO NO PUEDE DEFINIRSE SI PUEDE DESCRIBIRSE
- EL RESULTADO DEL DISEÑO NO PUEDE EXPERIMENTARSE
- SOLO PUEDE EXPERIMENTARSE CON LA IMPLEMENTACION RESULTANTE DEL DISEÑO
- LA CALIDAD DEL DISEÑO DETERMINA LA CALIDAD DEL PRODUCTO DE SOFTWARE CONSTRUIDO
- EL DISEÑO PUEDE EXPLICARSE

EXPLICACIONES DEL DISEÑO

- **DISEÑO COMO PROCESO DE RESOLUCION DE PROBLEMAS**
- **DEL PROBLEMA AL MODELO DE SOLUCION SE PASA POR TRES ESTADOS:**
  - **DIVERGENCIA**
  - **TRANSFORMACION**
  - **CONVERGENCIA**
- **EL DISEÑADOR DEBE IMAGINAR QUE EL SISTEMA FUE CONSTRUIDO Y ESTA SIENDO UTILIZADO PARA DECIDIR DETALLES**

- ***DISEÑO COMO PROBLEMA DEFECTUOSO***

SIRVE PARA ENTENDER EL ROL DEL DISEÑADOR Y CUALES SON SUS DESAFIOS

EL PROBLEMA NO PUEDE SER TOTALMENTE PLANTEADO, NO HAY REGLA NO HAY SOLUCION CORRECTA

- **HAY SOLUCIONES BUENAS O MALAS**
- **NO PUEDE SER TESTEADO EL DISEÑO**

EXPLICACIONES DEL DISEÑO:

•EL DISEÑO NO PUEDE EXPERIMENTARSE

•NO HAY SOLUCIONES LIMITADAS

•NO HAY FORMAS DE OBTENER SOLUCIONES LIMITADAS

•EL DISEÑO ES SINGULAR CON SUS PROPIEDADES

•HAY SINTOMAS DE PROBLEMAS DE ALTO NIVEL

•EL MODELO ES LA ESENCIA DE LA ACTIVIDAD

•EL DISEÑO COMO RESPUESTA FACTORES CRITICOS

•CONCORDANCIA DEL PROBLEMA CON LA SOLUCION

•AUMENTO DE LA COMUNICACIÓN CON EL USUARIO

•ENVOLTURA DEL DISEÑO

•PROBLEMA CRITICO: LIMITE DEL DISEÑO

•EJ. PROBLEMA DISEÑAR UN AEROPUERTO

•FACTOR CRITICO: TRANSPORTE DE GENTE O SEGURIDAD

Resumen

- El diseño es un proceso creativo
- Las actividades del diseño incluyen el diseño arquitectural, la especificación del sistema, el diseño de componentes, el diseño de la estructura de datos y el diseño de los algoritmos
- La descomposición funcional considera al sistema como un conjunto de unidades funcionales
- La descomposición orientada a objetos considera al sistema como un conjunto de objetos