

PARTE III

FASE DE DISEÑO EN COMET

DSIE (UPCT)

1

Contenido

- PARTE III: Fase de Diseño en COMET
 - ✦ Arquitecturas Software
 - ✦ Diseño Arquitectónico de Aplicaciones Distribuidas
 - ✦ Estructuración en Tareas
 - ✦ Diseño de Clases
 - ✦ Diseño Detallado del Software
 - ✦ Análisis de Productividad de Diseños Sw Concurrentes y de Tiempo Real

DSIE (UPCT)

2

Arquitecturas Software

DSIE (UPCT)

3

Arquitecturas Software

- "Divide y vencerás"
- Manejar con éxito la complejidad inherente a los sistemas grandes necesita de **mecanismos de descomposición de la arquitectura** del sistema.

DSIE (UPCT)

4

Estilos de Arquitecturas Sw

- **Patrones arquitectónicos** para sistemas concurrentes, de tiempo real y distribuidos.
- Estilo Arquitectónico Cliente/Servidor.
- Estilo Arquitectónico Multicapa.
- Estilo Arquitectónico de Tareas Comunicantes.
- Mezcla de estilos.

DSIE (UPCT)

5

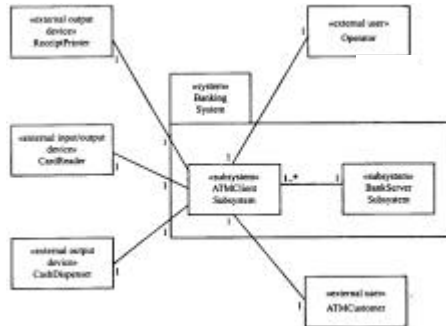
Arquitectura Cliente/Servidor

- Estilo Arquitectónico Cliente/Servidor:
 - ✦ Ampliamente usada.
 - ✦ Uno o más servidores proveen servicios a uno o más clientes.
 - ✦ Presente en la mayoría de arquitecturas distribuidas.
 - ✦ A veces, objetos *brokers* hacen de intermediarios ganando en transparencia e independencia de la plataforma.

DSIE (UPCT)

6

Arquitectura Cliente/Servidor

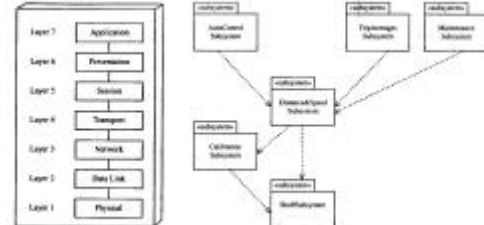


DSIE (UPCT)

7

Arquitectura Multicapa

- Los SOs, los SGBDs o el Sw de Comunicaciones son ejemplos de arquitectura multicapa.
- Usada también en el estándar OSI de ISO.



DSIE (UPCT)

Arquitectura de Tareas Multicapa

- Un conjunto de tareas que se comunican entre sí.
- Con o sin memoria compartida.
- Con memoria compartida, todas las tareas al mismo nodo computacional.

DSIE (UPCT)

9

Mezcla de estilos

- Los estilos no son excluyentes entre sí.
- Puede haber una arquitectura multicapa implementada como una red de tareas concurrentes.

DSIE (UPCT)

10

De Sistemas a Subsistemas

- Un subsistema contiene objetos funcionalmente dependiente entre ellos.
- Los sistemas deben tener bajo acoplamiento entre sí.
- Un subsistema es un objeto agregado que contiene a otros objetos que lo forman.

DSIE (UPCT)

11

De Sistemas a Subsistemas

- Se empieza con los casos de uso. Los objetos del mismo caso de uso no están geográficamente distribuidos.
- El modelo de interacción dado por el caso de uso determina objetos relacionados entre sí.
- Estos objetos están muy acoplados entre sí y menos con el resto => **subsistema**.
- Si el objeto participa en más de un caso de uso entonces se ubica en el que está más acoplado.

DSIE (UPCT)

12

D. de Colaboración
consolidados

- El **D. de Colaboración consolidado** es una síntesis de los d. de Colaboración de cada caso de uso.
- Para simplificar no se muestra la secuencia de numeración.
- COMET enfatiza el modelo dinámico**, crucial en los sistemas concurrentes,...
- El orden de síntesis puede ser el de ejecución de los casos de uso.
- Añadimos nuevos objetos, nuevos mensajes,...

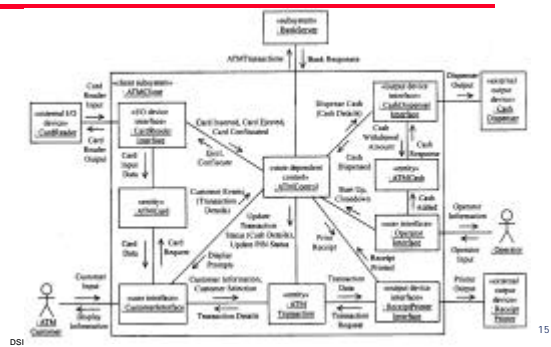
13

D. de Colaboración consolidados

- Se consideran las alternativas.
- Por simplificar se pueden agregar los mensajes.
- Se necesita un [diccionario](#).

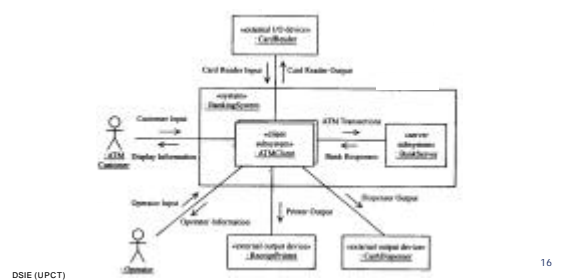
14

Ejemplo D. de Colaboración consolidado



D. de Colaboración de Subsistemas

- Es importante mostrar la relación estructural entre subsistemas en el **D. de Colab. de Subsistemas**.



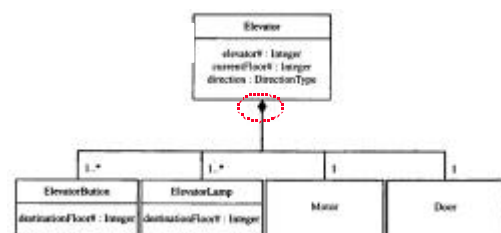
Guías de separación

- Para asegurar un alto acoplamiento dentro de un subsistema y un bajo acoplamiento entre subsistemas es importante considerar una serie de **guías de separación** de aspectos.
- **Objeto Agregado/Compuesto.**
 - ✦ Objetos relacionados por agregación deben ir al mismo subsistema.
 - ✦ Si es composición esta cohesión es mayor.

17

Guías de separación

- Ej. de agregado:



18

Localización Geográfica.

-

19

Letter 6 to Haver's

- DSIE (UPCT)

20

- DSIE (UPCT)

21

El tipo de los subsistemas suele ser dependiente de

- DSIE (UPCT)

23

- DSIE (UPCT)

23



2.

Subsistema Coordinador

- Necesario para coordinar la existencia de subsistemas de control dependientes entre sí.



DSIE (UPCT)

25

Subsistema de Colección de Datos

- Recoge datos del entorno.
- Los almacena y tras analizarlos y procesarlos los devuelve al entorno.



DSIE (UPCT)

26

Subsistema de Análisis de Datos

- Analiza una serie de datos y proporciona informes de los resultados a otros subsistemas.

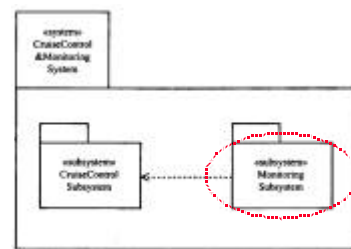


DSIE (UPCT)

27

Subsistema de Análisis de Datos

- Puede combinarse con colección de datos:



DSIE (UPCT)

28

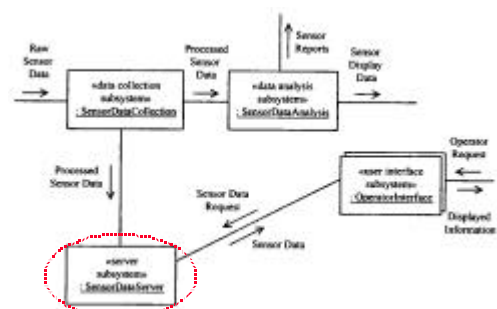
Subsistema Servidor

- Proporciona un servicio a otros subsistemas.
- Responde a peticiones de subsistemas cliente.
- No toma iniciativas.
- Puede ser muy simple con un solo objeto entidad o integrar de todo.
- Suelen ligarse a bases de datos y dispositivos de I/O.

DSIE (UPCT)

29

Subsistema Servidor



DSIE (UPCT)

30

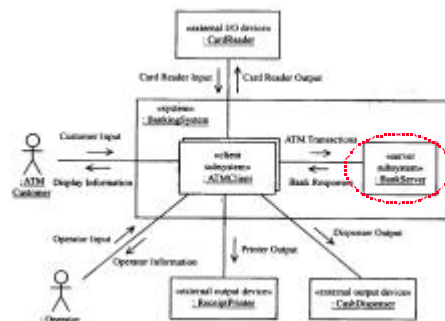
Subsistema Servidor



DSIE (UPCT)

31

Subsistema Servidor



DSIE (UPCT)

32

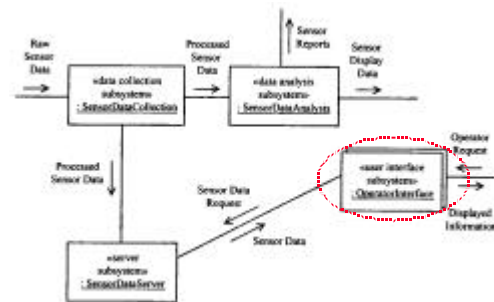
Subsistema de Interfaz de Usuario

- Provee un interfaz de usuario.
- Actúa como **cliente** proporcionando acceso a los usuarios a servicios provistos por uno o más servidores.
- Un subsistema por tipo de usuario.

DSIE (UPCT)

33

Subsistema de Interfaz de Usuario



DSIE (UPCT)

34

Subsistema de I/O

- Es habitual agrupar todas las clases de interfaz a dispositivos en un subsistema.
- Un número reducido de desarrolladores se dedican a este tipo de subsistemas.

DSIE (UPCT)

35

Subsistema Servicios de Sistemas

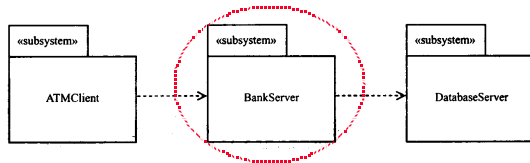
- Ciertos servicios no son propios del dominio del problema.
- La gestión de archivos o de las comunicaciones en red...
- Son usados por el resto de subsistemas.
- No suelen desarrollarse como parte de la aplicación.

DSIE (UPCT)

36

Ejemplos

- "Banking System" es un caso de arquitectura cliente/servidor jerarquizada en tres capas.

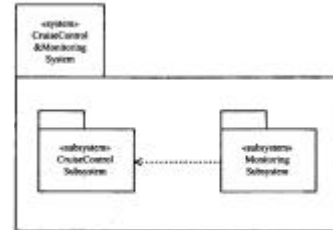


DSIE (UPCT)

37

Ejemplos

- El ejemplo del control de velocidad posee dos subsistemas débilmente acoplados.



DSIE (UPCT)

38

Ejemplos

- El subs. "cruise control subsystem" es de control.
- El subs. "monitoring subsystem" provee colección de datos y análisis.
- El único interfaz es el dato cumulative distance.

DSIE (UPCT)

39

Modelado Estático en el Diseño

- En el análisis se desarrolló un modelo estático del dominio del problema.
- Se representó en un diagrama de clases.
- En el diseño se da un modelo estático detallado en el **dominio de la solución**.

DSIE (UPCT)

40

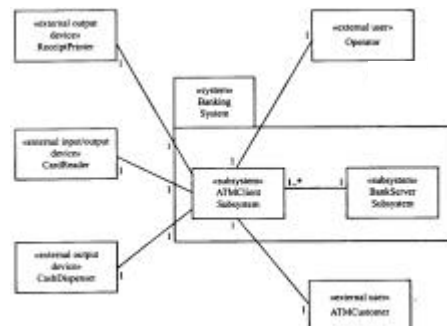
Modelado Estático en el Diseño

- Si el modelo estático del análisis está suficientemente detallado puede servir de base.
- Si es demasiado abstracto mejor partir del **D. de Colaboración Consolidado**:
 - ✳ Objetos instancias de clases.
 - ✳ Un enlace entre objetos => una relación entre clases.
- Se considera la dirección de navegación de las asociaciones.

DSIE (UPCT)

41

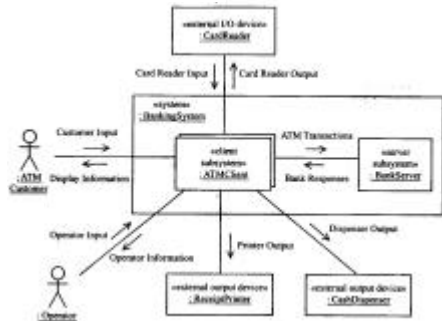
Ejemplo



DSIE (UPCT)

42

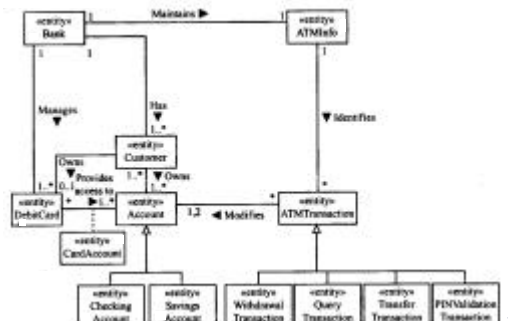
Ejemplo



DSIE (UPCT)

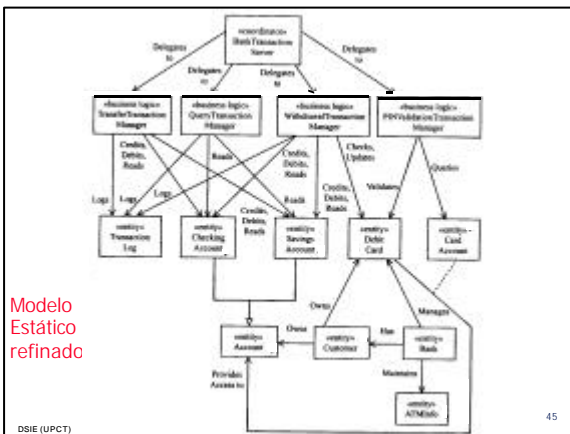
43

Ejemplo: Modelo Estático dominio del problema



DSIE I

44

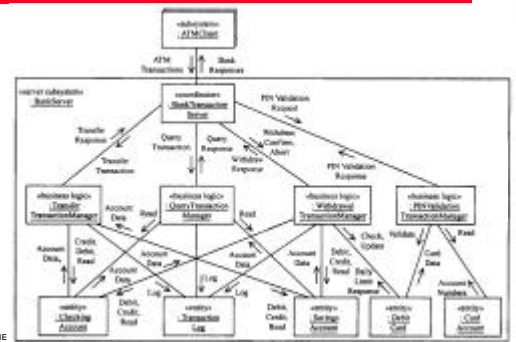


Modelo Estático refinado

DSIE (UPCT)

45

Ejemplo considerando el D. de Colab. consolidado



DSIE

16

Conclusión

- Para manejar con éxito la complejidad inherente a los sistemas Sw grandes es necesario tener un enfoque para descomponer el sistema en subsistemas y desarrollar la arquitectura Sw del sistema usando criterios de estructuración apropiados.
- Tras esto, cada subsistema podrá desarrollarse independientemente.

DSIE (UPCT)

47

Diseño Arquitectónico de Aplicaciones Distribuidas

DSIE (UPCT)

48

Diseño Arquitectónico de Aplicaciones Distribuidas

- Aplicaciones que ejecutan en nodos distribuidos geográficamente mediante una red de comunicaciones:
 - ✦ Aplicaciones cliente/servidor.
 - ✦ Aplicaciones de control distribuidas y de tiempo real.
 - ✦ Aplicaciones de colección de datos distribuidas y de tiempo real.
 - ✦ ...

DSIE (UPCT)

49

Diseño Arquitectónico de Aplicaciones Distribuidas

- Aplicación distribuida = subsistemas distribuidos.
- Subsistema
 - = componente configurable en un nodo lógico.
 - = colección de tareas concurrentes en un nodo lógico.
- En COMET se asume la existencia de acceso distribuido a los datos (si es necesario).

DSIE (UPCT)

50

Arquitecturas configurables y Componentes Software

- Objetivo: alta reconfigurabilidad.
- La misma arquitectura Sw adaptable a varias configuraciones de sistema.
- La asociación de subsistemas a nodos físicos no se hace en tiempo de diseño sino en la configuración del sistema.

DSIE (UPCT)

51

Arquitecturas configurables y Componentes Software

- La comunicación entre tareas de subsistemas distintos debe hacerse mediante comunicación por mensajes.
- Un componente distribuido es un objeto activo con un interfaz bien definido.

DSIE (UPCT)

52

Arquitecturas configurables y Componentes Software

- Un tipo de componentes es auto-contenido: puede compilarse, almacenarse y ser instanciado y linkado en una aplicación.
- Los tipos de componentes bien diseñados pueden utilizarse en aplicaciones distintas, más allá de para las que fueron diseñados inicialmente.

DSIE (UPCT)

53

Pasos en el Diseño de Aplicaciones Distribuidas

- Tres pasos principales:
 1. **Descomposición del Sistema:** en subsistemas separados que se comunican mediante mensajes.
 2. **Descomposición de Subsistemas:** estructurando los subsistemas en tareas concurrentes. Cada subsistema puede diseñarse usando algún método para sistemas no distribuidos.
 3. **Configuración del Sistema:** tras el diseño distribuido las instancias pueden definirse y configurarse. Estas instancias se definen, interconectan y asocian a la configuración Hw formada por nodos físicos.

DSIE (UPCT)

54

Descomposición del Sistema

- Partimos de los D. de Colaboración que muestran la comunicación entre objetos.
- Objetos con comunicación frecuente son candidatos al mismo subsistema.
- Un objeto sólo puede pertenecer a un subsistema.

DSIE (UPCT)

55

Descomposición del Sistema

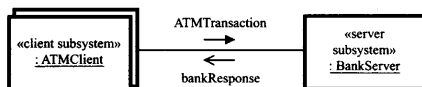
- La distribución geográfica determina en parte el particionamiento.
- Clientes suelen estar separados de servidores.

DSIE (UPCT)

56

Descomposición del Sistema

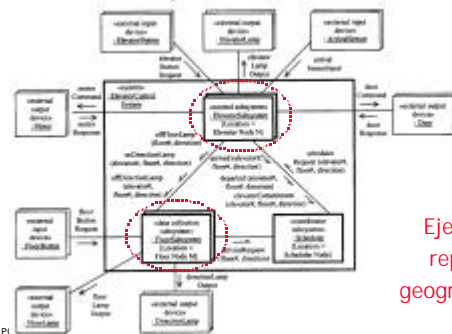
- Ejemplo de cliente/servidor:



DSIE (UPCT)

57

Descomposición del Sistema



Ejemplo
reparto
geográfico

DSIE (UPCT)

58

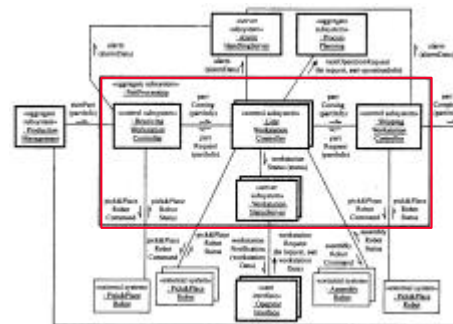
Subsistemas compuestos y agregados

- Un **subsistema compuesto** es un componente cuyos objetos o subsistemas **pertenecen a la misma localización geográfica**.
CONTENCIÓN FÍSICA y LÓGICA
- Un **subsistema agregado** es uno que agrupa objetos o subsistemas con funcionalidad similar o pertenecen al mismo caso de uso pero que **pueden estar repartidos geográficamente**.
CONTENCIÓN LÓGICA

DSIE (UPCT)

59

Subsistemas agregados



DSIE (UPCT)

60

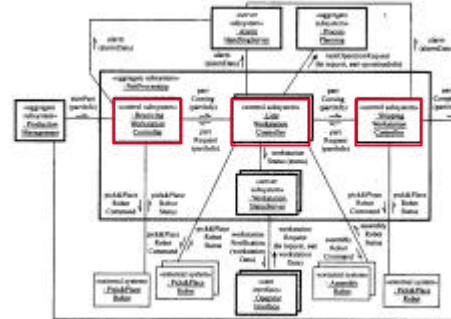
Subsistemas compuestos

- Es un componente que encapsula los objetos que contiene.
- No añade funcionalidad adicional.
- Los mensajes entrantes se encaminan a la tarea destino apropiada.
- Idem con las salidas.
- Los mecanismos de encaminamiento son dependientes de la implementación.

DSIE (UPCT)

61

Subsistemas compuestos



DSIE (UPCT)

62

Distribuido y configurable

- Se necesitan criterios que garanticen que los subsistemas encontrados puedan diseñarse como componentes distribuidos y asociados a nodos distribuidos.
- El paso de subsistema a nodo se hace en tiempo de configuración.
- Pero los subsistemas en diseño deben ser configurables...**

DSIE (UPCT)

63

Criterios para la Configuración de Componentes Distribuidos

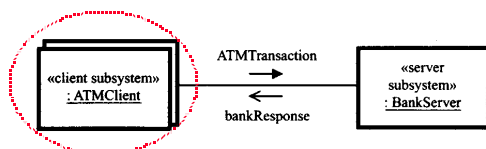
- En tiempo real y/o distribuido, un servicio provisto por un subsistema debe asociarse con una localización física o restringirse a ejecutarse en un recurso Hw específico.
- Proximidad** a las fuentes de datos físicos: acceso rápido...
- Autonomía** localizada. El componente controla un aspecto dado del sistema y su funcionamiento es relativamente independiente del resto.

DSIE (UPCT)

64

Criterios para la Configuración de Componentes Distribuidos

- Proximidad y autonomía se cumplen en:



DSIE (UPCT)

65

Criterios para la Configuración de Componentes Distribuidos

- y en:

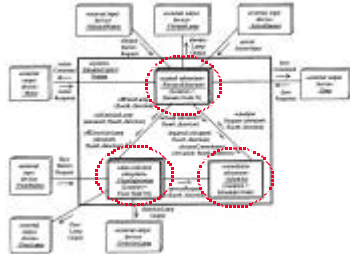


DSIE (UPCT)

66

Criterios para la Configuración de Componentes Distribuidos

- **Productividad**. En aplicaciones distribuidas los componentes de tiempo real llevan a cabo servicios críticos en un nodo específico.

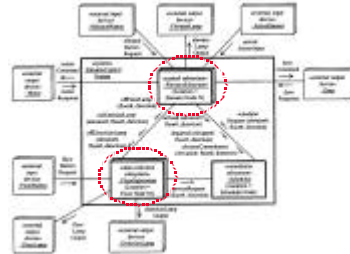


DSIE (UPCT)

67

Criterios para la Configuración de Componentes Distribuidos

- **Hw especializado**. El Hw puede determinar que un componente resida en un nodo dado o bien se conecte a dispositivos específicos. P.e. procesamiento vectorial, sensores,...

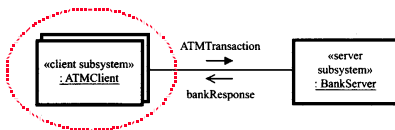


DSIE (UPCT)

68

Criterios para la Configuración de Componentes Distribuidos

- **Interfaz de Usuario**. Un componente que provee interfaz de usuario puede necesitar residir en un nodo local. Conexiones típicas a impresoras, pantallas...

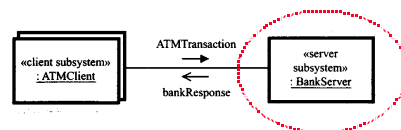


DSIE (UPCT)

69

Criterios para la Configuración de Componentes Distribuidos

- **Servidor**. Los servidores suelen proveer servicios asociados a repositorios de datos o bien relativos a dispositivos I/O locales. Es habitual que un servidor tenga su propio nodo.



DSIE (UPCT)

70

Criterios para la Configuración de Componentes Distribuidos

- o en ...



DSIE (UPCT)

71

Diseño de Interfaces de Subsistemas

- Toda la comunicación entre componentes es mediante **mensajes**.
- Comunicación por mensajes **asíncrona** o **síncrona**.

DSIE (UPCT)

72

Interfaz asíncrona

- Comunicación por mensajes **asíncrona**:
 - Colas FIFO o prioridades.
 - Mayor flexibilidad.
 - Destinatarios pueden ser grupos de objetos.
 - Ideal si el cliente no necesita respuesta o confirmación.
 - En ambiente distribuido el cliente recibe notificación de la llegada del mensaje al nodo destino.
 - Suele asociarse **timeout** en la confirmación de recepción del mensaje.
 - El concepto de transacción es más difícil de implementar.

DSIE (UPCT)

73

Interfaz asíncrona



DSIE (UPCT)

74

Interfaz síncrona

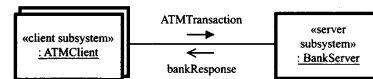
- Comunicación por mensajes **síncrona**:
 - Típica de clientes/servidor.
 - El cliente espera respuesta.
 - La delegación entre servidores es posible.
 - Puede necesitarse una cola en el servidor si hay muchos clientes.
 - El cliente manda un mensaje y espera la respuesta.
 - La respuesta puede ser que no llegó la solicitud.
 - Soportada por RPC y por RMI.
 - Puede establecerse una conexión duradera sobre la que comunicarse.
 - La variante síncrona sin respuesta en un ambiente distribuido no tiene demasiado sentido.

DSIE (UPCT)

75

Interfaz síncrona

- Un servidor y muchos clientes.
- Se necesita cola FIFO para atender las peticiones.



DSIE (UPCT)

76

Subscripción/Notificación

- Comunicación 1 a muchos necesaria en entornos distribuidos.
- También llamada **Comunicación Multicast**:
 - Las tareas se subscriben a un grupo y reciben mensajes destinados a miembros del grupo.
 - Una tarea puede borrarse de la lista.
 - Típica en Internet.

DSIE (UPCT)

77

Subscripción/Notificación



DSIE (UPCT)

78

Comunicación en BroadCast

- Un mensaje que no ha sido solicitado previamente se envía a un conjunto de destinatarios.
- Por ejemplo la notificación de un "shutdown".
- Cada destinatario decide si procesar la petición o rechazarla.

DSIE (UPCT)

79

Comunicación mediante Brokers

- En entornos distribuidos, clientes y servidores son objetos distribuidos.
- Un **objeto broker** es un intermediario entre cliente y servidores.
- Libera a los clientes de saber quién y dónde se proveen determinados servicios.
- Provee además **transparencia de localización**: si el servidor se *mueve* sólo el broker debe ser notificado.

DSIE (UPCT)

80

Comunicación mediante Brokers

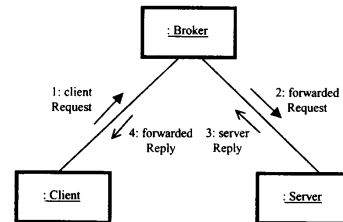
- Mediante el **servicio de nombres** el broker se entera de los servicios previstos por cuáles servidores.
- El broker recibe la respuesta del servidor y la encamina al cliente.

DSIE (UPCT)

81

Brokers

- Páginas blancas con encaminamiento:

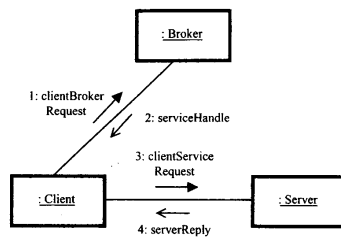


DSIE (UPCT)

82

Brokers

- Páginas blancas con manejador:

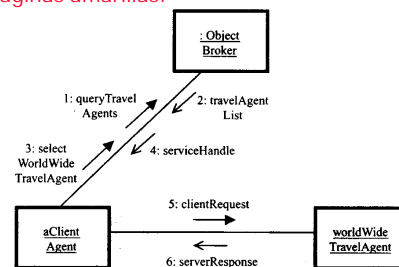


DSIE (UPCT)

83

Brokers

- Páginas amarillas:



DSIE (UPCT)

84

Diseño de Subsistemas Servidores

- Los **subsistemas servidores** desempeñan un papel importante en el diseño de aplicaciones distribuidas.
- Un subsistema servidor provee un servicio para subsistemas clientes.
- Típicos: servidores de ficheros, de bases de datos, de líneas de impresión...

DSIE (UPCT)

85

Diseño de Subsistemas Servidores

- En sistemas distribuidos, el acceso a objetos pasivos pasa por ubicarlos en algún servidor.
- El servidor provee el servicio para el acceso a estos objetos pasivos.
- Dos tipos: **secuenciales** y **concurrentes**.

DSIE (UPCT)

86

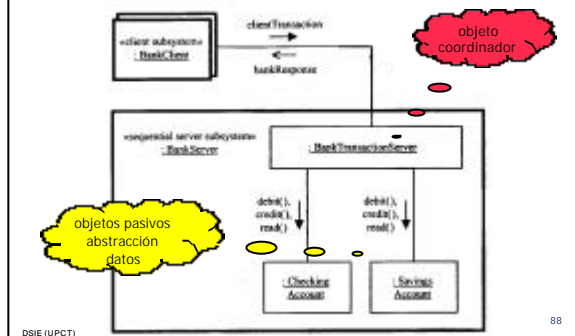
Subsistemas Servidores Secuenciales

- Sirven peticiones de clientes de manera secuencial.
- Se diseñan como una única tarea.
- El objeto coordinador del servidor desempaqueta el mensaje y, según el tipo de mensaje, invoca la operación apropiada del objeto servidor.
- La respuesta se empaqueta y se envía de vuelta al cliente.

DSIE (UPCT)

87

Subsistemas Servidores Secuenciales



DSIE (UPCT)

88

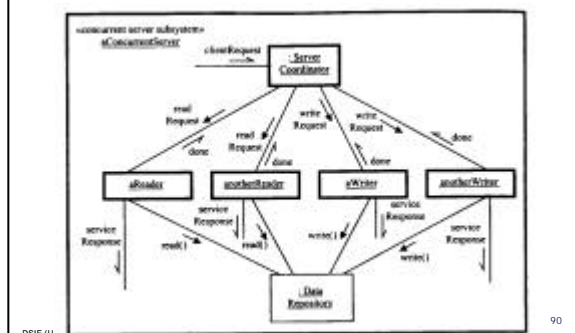
Subsistemas Servidores Concurrentes

- Si la demanda de servicios es elevada el que el servidor sea secuencial puede suponer un **cuello de botella** en el sistema.
- Un acceso concurrente a los datos con varias tareas es la solución.
- Se necesita **sincronización** en el acceso a los datos compartidos:
 - Exclusión mutua.
 - Múltiples lectores y escritores.

DSIE (UPCT)

89

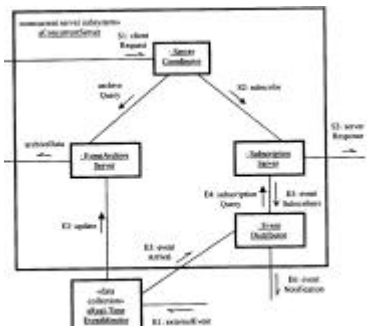
Subs. Serv. Concurrentes mediante lectores/escritores



DSIE (U)

90

Subs. Serv. Concurrentes mediante subscripción/notificación



DSIE (UPCT)

91

Distribución de Datos

- Los servidores secuenciales y concurrentes poseen repositorios de datos **centralizados**.
- El **servidor** puede llegar a ser un **cuello de botella**.
- Dos enfoques posibles:
 - Servidores Distribuidos.**
 - Replicación de Datos.**

DSIE (UPCT)

92

Servidores Distribuidos

- Varias localizaciones para los datos.
- Cada localización posee un servidor local que responde a peticiones sobre los datos de dicha localización.

DSIE (UPCT)

93

Ejemplo de Servidores Distribuidos



DSIE (UPCT)

94

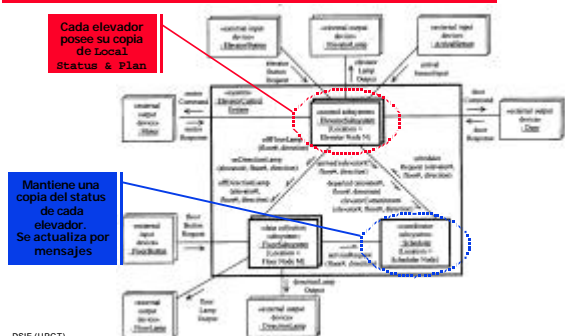
Replicación de Datos

- El mismo dato duplicado en más de una localización.
- Se acelera el acceso a los datos.
- Debe haber procedimientos que aseguren la actualización de los datos correctamente.

DSIE (UPCT)

95

Ejemplo de Replicación de Datos



DSIE (UPCT)

Configuración del Sistema

- Después de diseñar una aplicación distribuida hay que crear **instancias** y **configurarlas**.
- Instancia = sistema destino.
- Configuración distribuida = múltiples nodos distribuidos geográficamente y conectados por red.

DSIE (UPCT)

97

Aspectos de la Config. de Sistemas

- ¿Qué instancias crear?
- ¿Cómo conectar las instancias?
- ¿Cómo deben localizarse en los nodos?

DSIE (UPCT)

98

Actividades de la Config. de Sistemas

- Definición de instancias de los tipos de componentes.
 - Número de elevadores, núm. de pisos.
 - Una instancia `Elevator Subsystem` por elevador.
 - Una instancia `Floor Subsystem` por piso.
 - Nombre únicos por elevador y piso.
- Interconexión de las instancias de componente.
 - Elevadores, pisos y el planificador intercambian mensajes entre sí sabiendo los identificadores de cada uno de ellos: **interconexión previa**.

DSIE (UPCT)

99

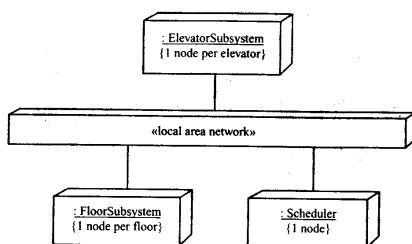
Actividades de la Config. de Sistemas

- Las instancias de componentes (nodos lógicos) se pasan a nodos físicos.
 - Por ejemplo, decidir que dos componentes van al mismo nodo físico o a distinto.

DSIE (UPCT)

100

Ejemplo de Configuración del Sistema



DSIE (UPCT)

101

Más...

- Tras determinar los subsistemas de la aplicación distribuida y el interfaz de mensajes entre sí, el siguiente paso es diseñar cada subsistema.
- Cada subsistema consiste de una o más tareas concurrentes.
- En principio, todas las tareas del mismo subsistema se ejecutarán en el mismo nodo.
- Las tareas del mismo nodo pueden comunicarse, mediante asincronía, o vía objetos pasivos que residan en memoria compartida.

DSIE (UPCT)

102

Estructuración en Tareas

DSIE (UPCT)

103

Introducción

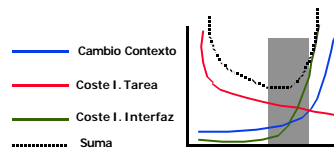
- Una **tarea** es un objeto activo que tiene su propio hilo de ejecución.
- Una **arquitectura de tareas** descompone un sistema en tareas más interfaces más interconexiones.
- Los **criterios de estructuración en tareas** permiten mediante guías determinar las tareas que conforman un subsistema.

DSIE (UPCT)

104

Introducción

- Las tareas podrán ser procesos ligeros o pesados, según el sistema.
- Demasiadas tareas significa demasiado cambio de contexto



DSIE (UPCT)

105

Introducción

- El diseñador debe buscar el compromiso:
 - ✦ Muchas tareas facilitan la claridad y el diseño, pero no demasiadas...
- ¿Qué objetos **pueden** ejecutar concurrentemente con otros?
- ¿Qué objetos **deben** ejecutar concurrentemente con otros?

DSIE (UPCT)

106

Introducción

- Objetos restringidos a ejecutar secuencialmente son agrupados en la misma tarea.
- Una tarea puede incluir a más de un objeto.

DSIE (UPCT)

107

Categorías de Estructuración para Tareas

- Los criterios de estructuración de tareas se agrupan por categorías.
- **Criterio de est. de tareas I/O.** Cómo los objetos de interfaz a dispositivos se pasan a tareas I/O y cuándo se activan.
- **Criterio de est. de tareas internas.** Cómo los objetos internos se pasan a tareas internas y cuándo se activan.

DSIE (UPCT)

108

Categorías de Estructuración para Tareas

- Criterio de **prioridad** de tareas. Analiza la importancia de ejecutar una tarea con cierta prioridad respecto de otras.
- Criterio de **agrupación** de tareas. Cómo los objetos pueden agruparse en tareas concurrentes.
- Criterio de **inversión** de tareas. Cómo puede reducirse el número de tareas si es preciso.

DSIE (UPCT)

109

Categorías de Estructuración para Tareas

- Las tareas además pueden ser **periódicas** o **aperiódicas**.
- Los criterios anteriores se desarrollan en dos fases:
 - ✦ 1ª fase: I/O, Internas y Prioridades.
 - ✦ 2ª fase: Agrupación e Inversión.

DSIE (UPCT)

110

Categorías de Estructuración para Tareas

- Criterio de est. de tareas **I/O**.
- Criterio de est. de tareas **internas**.
- Criterio de **prioridad** de tareas.
- Criterio de **agrupación** de tareas.
- Criterio de **inversión** de tareas.

DSIE (UPCT)

111

Características dispositivos I/O

- Las características de los dispositivos I/O a los que se conectan las tareas no forman parte del análisis.
- Pero es esencial para determinar las propiedades de las tareas...
- Es necesario conocer las características Hw de los dispositivos I/O y la naturaleza de los datos de I/O al sistema.

DSIE (UPCT)

112

Características dispositivos I/O

- Características de dispositivos I/O:
 - ASÍNCRONOS (o activos)**:
(input) generan una interrupción cuando han generado un dato que debe ser procesado por el sistema.
(output) generan una interrupción cuando ya han procesado el dato recibido desde el sistema.
 - PASIVOS**:
No hay interrupciones. Se lee o escribe periódicamente o a demanda. Si es periódico, ¿con qué frecuencia?
 - ENLACE DE COMUNICACIONES**:
Un protocolo define cómo comunicarse (TCP/IP...). Las tareas en el nivel de aplicación se comunican mediante mensajes...

DSIE (UPCT)

113

Características dispositivos I/O

- Características de los datos:
 - ¿Se proveen datos **discretos** (finitos) o **analógicos** (continuos)?
 - Si es analógico lo habitual es leer a **demanda**. Por interrupciones sería inviable dado lo infinitésimo del cambio.

DSIE (UPCT)

114

Tipos de Tareas para dispositivos I/O

- Abordaremos:
 - Tareas de Interfaz a Dispositivo Asíncrono I/O.
 - Tareas de Interfaz a Dispositivo Periódico I/O.
 - Tareas de Interfaz a Dispositivo Pasivo.
 - Tareas Monitoras de Recursos.

DSIE (UPCT)

115

Tareas de Interfaz a Dispositivo I/O Asíncrono

- Por cada dispositivo asíncrono I/O se necesita una **tarea de interfaz a dispositivo asíncrono I/O**.
- La tarea se activa por interrupciones (aperiódica).
- Cada objeto de interfaz a dispositivo de este tipo se traslada a la tarea indicada.

DSIE (UPCT)

116

Tareas de Interfaz a Dispositivo I/O Asíncrono

- La tarea debe ejecutarse a la velocidad del dispositivo I/O con el que interacciona.
- La tarea puede tener que esperar indefinidamente la entrada pero si llega la interrupción el dato debe ser procesado para evitar pérdidas.
- El procesamiento puede hacerlo otra tarea.

DSIE (UPCT)

117

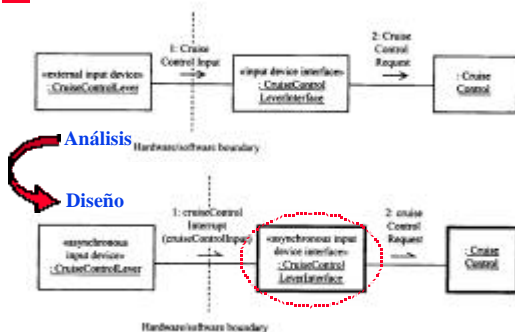
Tareas de Interfaz a Dispositivo I/O Asíncrono

- El dispositivo asíncrono puede ser un **sistema externo**. En este caso se intercambian mensajes.

DSIE (UPCT)

118

Tareas de Interfaz a Dispositivo I/O Asíncrono



DSI

119

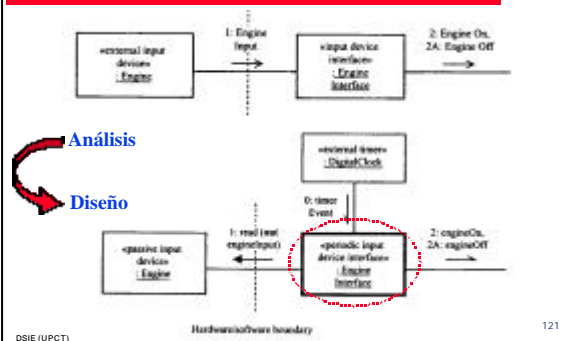
Tareas de Interfaz a Dispositivo I/O Periódico

- El dispositivo I/O es **pasivo** y se conecta a una **tarea de interfaz a dispositivo periódico**.
- La tarea es activada por un **timer**, efectúa la operación y espera al siguiente evento.
- El **periodo de la tarea** es el intervalo entre activaciones.
- Habitual con **sensores analógicos y digitales** que deben ser muestreados periódicamente.

DSIE (UPCT)

120

Tareas de Interfaz a Dispositivo I/O Periódico



Tareas de Interfaz a Dispositivo I/O Periódico

- La frecuencia con la que la tarea muestrea el dispositivo está en función de cómo se espera que cambien los valores y del retraso permitido en la lectura.
- De otro modo, la frecuencia depende de las características del dispositivo y de las propiedades del entorno de la aplicación.
- Lector de temperatura: frecuencia en minutos.
- Sensor de freno: frecuencia en milisegundos.
- Esta tarea es la adecuada para lecturas analógicas.

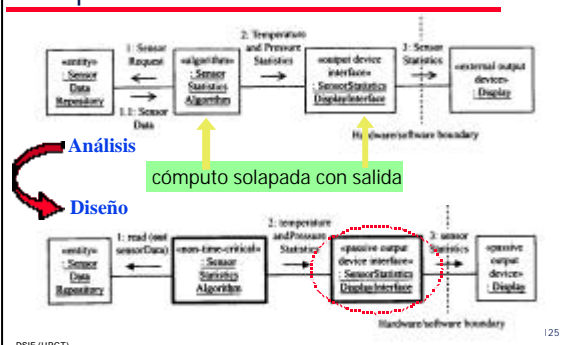
Tareas de Interfaz a Dispositivo I/O Pasivo

- Las **tareas de interfaz a dispositivo I/O pasivo** se usan cuando se tratan dispositivos de este tipo y que no necesitan tareas periódicas.
- Pasivo es el dispositivo; el objeto es activo (es una tarea...).
- Es habitual solapar acceso al dispositivo con cálculos.

Tareas de Interfaz a Dispositivo I/O Pasivo

- Si la computación se hace durante el acceso al dispositivo entonces lectura y computación van a tareas separadas.
- Si la computación necesita del dato de entrada entonces pueden hacerse en el mismo thread.
- Estos dispositivos suelen ser mayoritariamente de **salida** pues es más común solapar salida con cómputo.
- Cuando se solapa entrada y cómputo lo habitual es usar una tarea periódica de entrada.

Tareas de Interfaz a Dispositivo I/O Pasivo



Tareas Monitoras de Recursos

- Una tarea monitora de recursos es un caso particular de tarea pasiva I/O.
- Un dispositivo I/O que recibe peticiones de múltiples fuentes debería tener una tarea monitor para coordinar estas peticiones.
- Esto es necesario aunque el dispositivo sea pasivo.

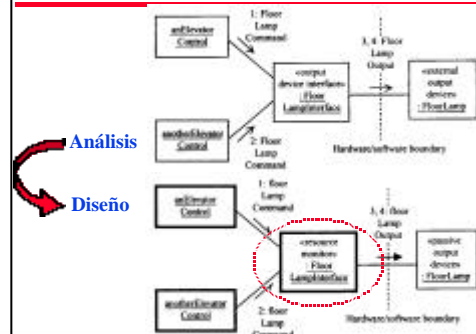
Tareas Monitoras de Recursos

- Objetivo: secuenciar las peticiones para garantizar la integridad y no se pierdan datos.
- P.e. si dos tareas escriben en una impresora simultáneamente puede salir cualquier cosa.

DSIE (UPCT)

127

Tareas Monitoras de Recursos



DSIE (UPCT)

128

Categorías de Estructuración para Tareas

- Criterio de est. de tareas **I/O**.
- Criterio de est. de tareas **internas**.
- Criterio de **prioridad** de tareas.
- Criterio de **agrupación** de tareas.
- Criterio de **inversión** de tareas.

DSIE (UPCT)

129

Criterios de Estructuración de Tareas Internas

- Distinguimos los siguientes tipos de tareas internas:
 - Tareas Periódicas.
 - Tareas Asíncronas.
 - Tareas de Control.
 - Tareas de Interfaz de Usuario.

DSIE (UPCT)

130

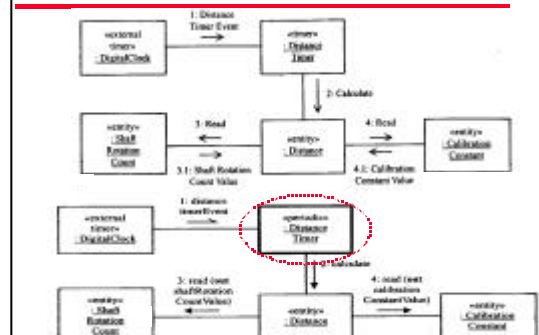
Tareas Internas Periódicas

- Es habitual en entornos de tiempo real y concurrentes la presencia de tareas periódicas correspondientes a actividades periódicas.
- Dos tipos:
 - Tareas periódicas algorítmicas.
 - Tareas periódicas de la lógica de negocios.
- Estas tareas son activadas por un **timer**, ejecutan su actividad y esperan el siguiente evento de activación.

DSIE (UPCT)

131

Tareas Internas Periódicas



DSIE

132

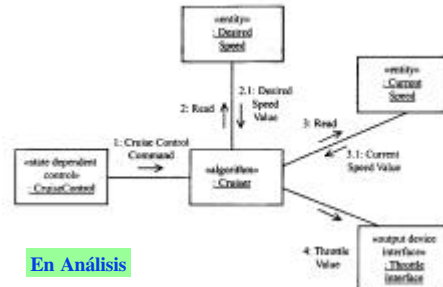
Tareas Internas Asíncronas

- En sistemas de tiempo real y concurrentes suele haber tareas que necesitan ejecutarse a demanda.
- Análogas a las asíncronas de dispositivo: el evento de activación esta vez es interno.
- Ante la activación se lleva a cabo una actividad y luego queda a la espera de la siguiente.
- Tipos: de la **lógica de negocios** o **algorítmicas**.

DSIE (UPCT)

133

Tareas Internas Asíncronas



DSIE (UPCT)

134

Tareas Internas Asíncronas



DSIE (UPCT)

135

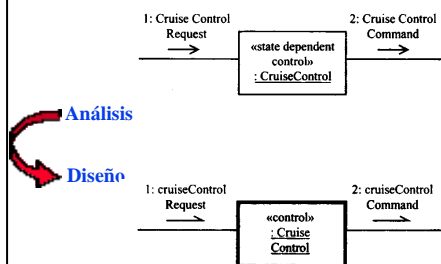
Tareas de Control

- COMET utiliza la variante restringida de *statechart* por la que no existe concurrencia intraobjetual explícita.
- Un statechart es secuencial y puede ejecutarse con una tarea: una **tarea de control**.
- Los objetos control coordinadores dan lugar a una **tarea coordinadora** controlando otras tareas y sin dependencia del estado.

DSIE (UPCT)

136

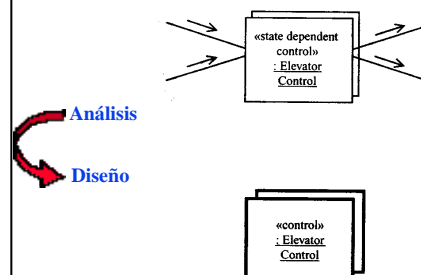
Tareas de Control



DSIE (UPCT)

137

Tareas de Control



DSIE (UPCT)

138

Tarea de Interfaz de Usuario

- La actividad de un usuario es un conjunto secuencial de operaciones que puede ser gestionada por una **tarea de interfaz de usuario**.
- Esta tarea interactúa con varios dispositivos típicos I/O manejados por el SO.
- Para los dispositivos estándar no es necesario incluir tareas de interfaz a dispositivo para manejarlas.

DSIE (UPCT)

139

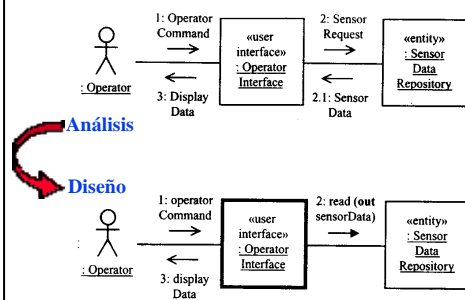
Tarea de Interfaz de Usuario

- En UNIX es típico una tarea por usuario.
- Todas las tareas del mismo usuario se ejecutan concurrentemente.
- En Windows una tarea por ventana...

DSIE (UPCT)

140

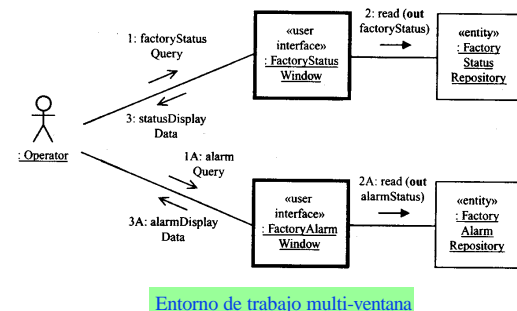
Tarea de Interfaz de Usuario



DSIE (UPCT)

141

Tarea de Interfaz de Usuario



DSIE (UPCT)

142

Categorías de Estructuración para Tareas

- Criterio de est. de tareas **I/O**.
- Criterio de est. de tareas **internas**.
- Criterio de **prioridad** de tareas.
- Criterio de **agrupación** de tareas.
- Criterio de **inversión** de tareas.

DSIE (UPCT)

143

Criterios de Prioridad de Tareas

- El criterio de prioridad de tareas considera la prioridad de las tareas a la hora de estructurarlas.
- Suele abordarse tarde en el ciclo de desarrollo.
- Es necesario para identificar objetos críticos en restricciones temporales o no que sean computacionalmente intensivos.

DSIE (UPCT)

144

Criterios de Prioridad de Tareas

■ Tareas críticas en tiempo:

- ✘ Es una tarea que necesita cumplir un plazo de manera estricta.
- ✘ Necesita una prioridad alta.
- ✘ Necesaria en la mayoría de sistemas en tiempo real.
- ✘ Muy ligadas a dispositivos asíncronos I/O.

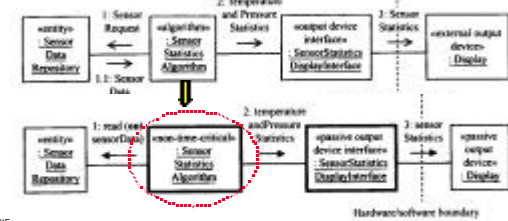
DSIE (UPCT)

145

Criterios de Prioridad de Tareas

■ Tareas computacionalmente intensivas pero no críticas en tiempo:

- ✘ Pueden ejecutarse como tareas de baja prioridad consumiendo ciclos de ocio de la CPU.



DSIE

146

Categorías de Estructuración para Tareas

- Criterio de est. de tareas **I/O**.
- Criterio de est. de tareas **internas**.
- Criterio de **prioridad** de tareas.
- Criterio de **agrupación** de tareas.
- Criterio de **inversión** de tareas.

DSIE (UPCT)

147

Criterios para Agrupación de Tareas

- En un sistema grande, muchos objetos puede significar muchas tareas.
- Según el modelo Actor [Agha 1986] esta relación es 1 a 1.
- Esto puede significar demasiada complejidad y sobrecarga del sistema.

DSIE (UPCT)

148

Criterios para Agrupación de Tareas

- El **criterio de Agrupación de Tareas** se usa para ver si es posible consolidar las tareas con vistas a reducir su número.
- También es referido como **criterio de cohesión de tareas**.
- Todas las tareas creadas por los criterios previos son candidatas a la agrupación.

DSIE (UPCT)

149

Criterios para Agrupación de Tareas

- Generalmente, si dos tareas están restringidas a no ejecutarse concurrentemente entonces deben ejecutar secuencialmente.
- Tipos de agrupación:
 - ▶ Temporal
 - ▶ Secuencial
 - ▶ De Control
 - ▶ De Exclusividad Mutua

DSIE (UPCT)

150

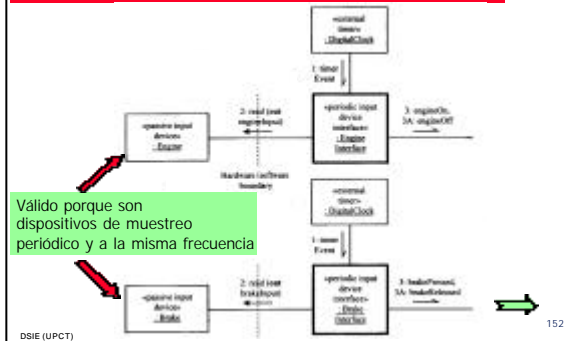
Agrupación Temporal

- Cuando no hay dependencia secuencial entre tareas y son activadas por el mismo evento entonces pueden ser agrupadas en una tarea.
- El orden lo decide el diseñador pero no influye en el resultado.
- Habitual en tareas periódicas con la misma frecuencia de activación.

DSIE (UPCT)

151

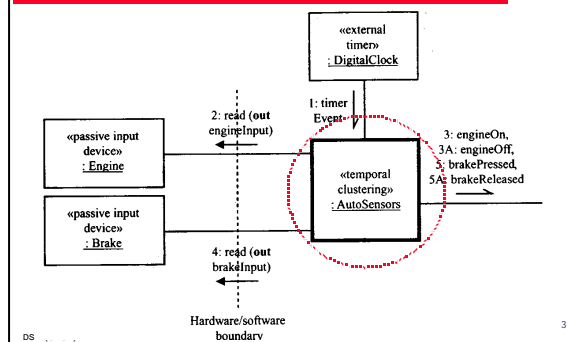
Agrupación Temporal



DSIE (UPCT)

152

Agrupación Temporal



DS

3

Consideraciones de la Agrupación Temporal

- No agrupar si una tarea es crítica y la otra no (pues darán prioridades distintas).
- No agrupar si está previsto se ejecuten en procesadores distintos.
- Preferencia para agrupar si están funcionalmente relacionadas y tienen la misma importancia desde el pdv de la planificación.
- Será posible agrupar si los periodos son proporcionales en cantidades enteras (p.e. 50 y 100).

DSIE (UPCT)

154

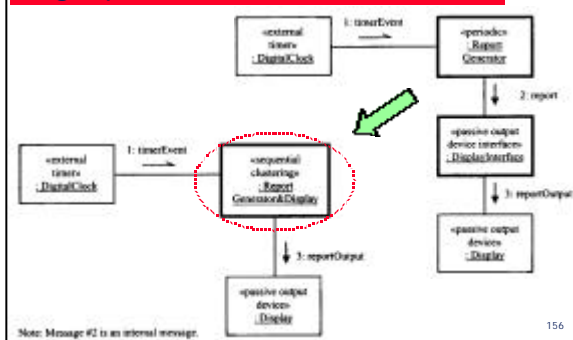
Agrupación Secuencial

- Se aplica cuando ciertas tareas están obligadas a ejecutarse secuencialmente.
- La primera tarea se activa asincrónamente o mediante evento periódico.
- Las siguientes se ejecutan secuencialmente después.

DSIE (UPCT)

155

Agrupación Secuencial



Note: Message #2 is an internal message.

156

Consideraciones de la Agrupación Secuencial

- La secuencia de tareas termina con una que no manda un mensaje inter-tareas.
- No incluir en la secuencia tareas que puedan recibir eventos de activación desde otras.
- Si sigue en la secuencia una tarea de baja prioridad de otra crítica se deben incluir como separadas.

157

Agrupación de Control

- Las tareas de control llevan a cabo **acciones** o habilitan **actividades** según el statechart definido.
- Acciones disparadas debido a una transición de estado:**
 - Suponemos es una operación de otro objeto.
 - Tiene por duración el tiempo de la transición de estado.
 - Se ejecuta dentro del thread del objeto control.
 - Si todas las acciones de dicho objeto se ejecutan dentro del mismo thread del objeto control entonces se combina con él.**

158

Agrupación de Control

- **Actividades habilitadas/deshabilitadas en una transición de estado:**
 - Suponemos es una actividad desarrollada por otro objeto.
 - Se habilita en la transición y deshabilita en otra transición posterior.
 - **Esta actividad debe estructurarse como una tarea aparte dado que se ejecuta concurrentemente al objeto control.**

159

Agrupación de Control

- **Actividades que se ejecutan durante un estado:**
 - Suponemos que el objeto control es quien la activa.
 - La actividad la ejecuta otro objeto.
 - Si la actividad es “activada” por el objeto control entonces se estructura como tarea aparte pues deben ejecutarse concurrentemente.
 - Si la actividad es la que detecta el cambio de estado y lo notifica mediante evento al objeto control y si este evento es el único que causa el cambio de estado la actividad puede agruparse con el objeto control.
 - Si el objeto control puede ser activado por otro evento de otra fuente entonces debe ser una tarea aparte.

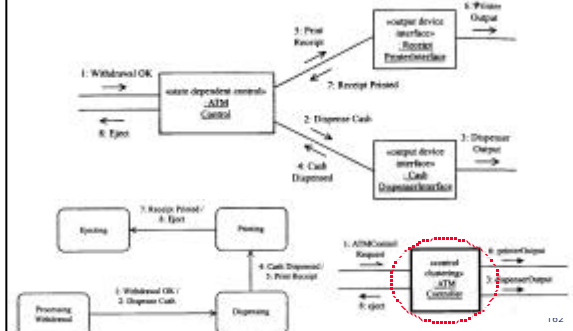
160

Agrupación de Control

- Un objeto origen manda mensajes al objeto control que causa un cambio de estado:
 - Si es la única causa de cambio de estado entonces puede agruparse con el objeto control según el criterio de agrupación secuencial.

161

Agrupación de Control



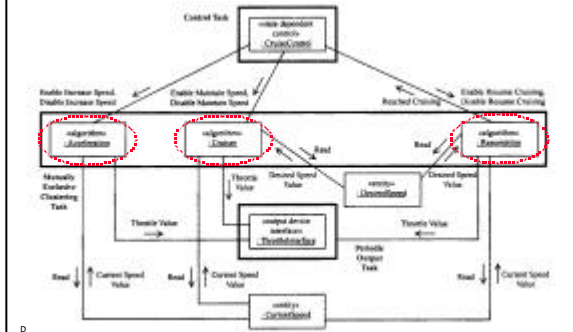
Agrupación de Exclusividad Mutua

- Cuando un grupo de tareas son **mutuamente exclusivas** entre sí pueden ser entonces agrupadas en una sola tarea.
- Se reduce el número de tareas.

DSIE (UPCT)

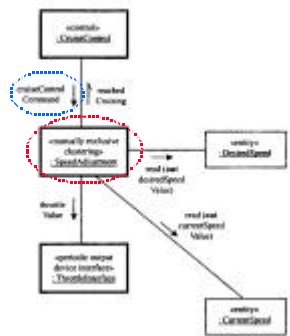
163

Agrupación de Exclusividad Mutua



D

Agrupación de Exclusividad Mutua



DSIE (UPCT)

165

Categorías de Estructuración para Tareas

- Criterio de est. de tareas **I/O**.
- Criterio de est. de tareas **internas**.
- Criterio de **prioridad** de tareas.
- Criterio de **agrupación** de tareas.
- Criterio de **inversión** de tareas.

DSIE (UPCT)

166

Criterio de Inversión de Tareas

- El concepto de inversión de tareas se remonta al año 1983 [M. Jackson] por el que el número de tareas de un sistema puede reducirse significativamente.
- El caso extremo lleva de una solución concurrente a otra secuencial.
- Suele aplicarse de manera temprana cuando se prevee una sobrecarga elevada del sistema.

DSIE (UPCT)

167

Criterio de Inversión de Tareas

- Tres formas de inversión:
 - ✦ Inversión de Tarea por Múltiple Instancia.
 - ✦ Inversión de Tareas Secuencial
 - ✦ Inversión de Tareas Temporal

DSIE (UPCT)

168

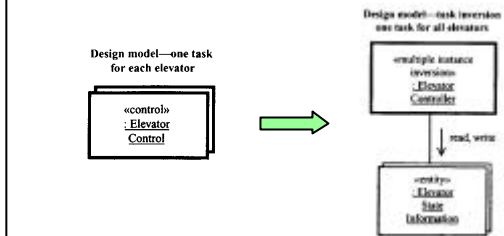
Inversión de Tarea por Múltiple Instancia

- Cuando hay muchos objetos control del mismo tipo se obtiene un número elevado de tareas.
- Con la **inversión** todas las tareas idénticas del mismo tipo se sustituyen por una que realiza el mismo servicio.
- La información del estado de cada objeto se mantiene en un objeto entidad pasivo separado.

DSIE (UPCT)

169

Inversión de Tarea por Múltiple Instancia



DSIE (UPCT)

170

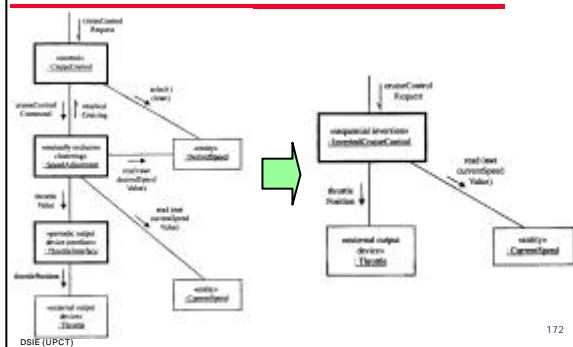
Inversión de Tareas Secuencial

- Se usa cuando hay acoplamiento elevado entre varias tareas.
- La combinación da lugar a llamada de operaciones entre las tareas en vez de mensajes.

DSIE (UPCT)

171

Inversión de Tareas Secuencial



DSIE (UPCT)

172

Inversión de Tareas Temporal

- Similar al agrupamiento temporal.
- Dos o más tareas periódicas (internas, I/O, de agrupación temporal) se combinan en una.
- La tarea tiene un procedimiento de planificación conducido por un *timer* que determina cuándo llamar a una operación específica.

DSIE (UPCT)

173

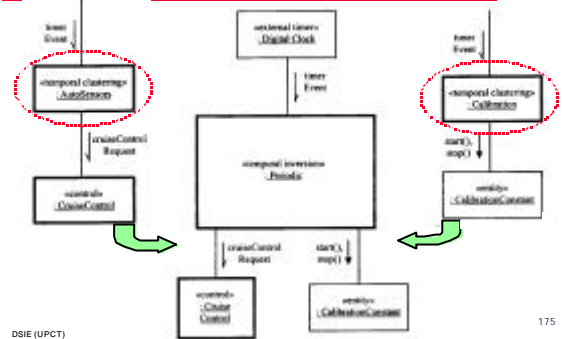
Inversión de Tareas Temporal

- Nótese que no están relacionados los objetos agrupados.
- Desde un pdv del diseño no es adecuado aplicar inversión temporal.
- Necesario para optimizar en situaciones donde la sobrecarga es elevada.

DSIE (UPCT)

174

Inversión de Tareas Temporales



DSIE (UPCT)

175

Desarrollo de la Arquitectura de Tareas

- Los criterios anteriores se aplican sobre el modelo de análisis tras decidir para cada objeto si es activo (tarea) o pasivo y en el siguiente orden:

1. Tareas de Interfaz de Dispositivo.
2. Tareas de Control.
3. Tareas Periódicas.
4. Otras tareas Internas.

DSIE (UPCT)

176

Desarrollo de la Arquitectura de Tareas

- Tareas de Interfaz de Dispositivo.**
 - Empezar con los objetos interfaz de dispositivo que interactúan con el mundo real.
 - Determinar si el objeto debe ser estructurado como un interfaz a dispositivo asíncrono I/O, un interfaz a dispositivo periódico I/O, un interfaz a dispositivo pasivo I/O, un monitor de recursos o una tarea periódica agrupada temporal.

DSIE (UPCT)

177

Desarrollo de la Arquitectura de Tareas

- Tareas de Control.**
 - Analizar cada objeto control dependiente del estado y estructurarlo como una tarea.
 - Cualquier objeto que ejecute una acción disparada desde el objeto control es candidato a combinarse con él.
 - Cualquier actividad habilitada debería estructurarse como una tarea independiente.
 - Usar el mismo enfoque para objetos control múltiples dependientes del estado. Ver si cada objeto puede ser llevado a una tarea o si debe aplicarse inversión para reducir la sobrecarga.

DSIE (UPCT)

178

Desarrollo de la Arquitectura de Tareas

- Tareas Periódicas.**
 - Analizar las actividades periódicas internas estructuradas como tareas periódicas.
 - Determinar si las tareas periódicas son disparadas por el mismo evento para entonces agrupar en la misma tarea (agrupación temporal).
 - Tareas que se hagan en secuencia pueden también ser agrupadas.

DSIE (UPCT)

179

Desarrollo de la Arquitectura de Tareas

- Otras Tareas Internas.**
 - Para cada tarea candidata interna activada por un evento interno identificar si alguna tarea adyacente del d. de colaboración concurrente puede agruparse en la misma tarea acorde al criterio de agrupación temporal, secuencial o de exclusividad.

DSIE (UPCT)

180

Resumen guías (1/2)

Analysis Model (object)	Design Model (task)
User interface	User interface Sequential clustering Control clustering
Device interface (input, output, I/O)	Asynchronous device interface (input, output, I/O) Periodic device interface (input, output, I/O) Passive device interface (usually output device interface) Resource monitor (usually output device interface) Temporal clustering (usually input device interface) Sequential clustering Control clustering (usually output device interface)
System interface	Asynchronous system interface Any clustering criterion
Entity	Sequential server Concurrent server Any clustering criterion

DSIE (UPCT)

181

Resumen guías (2/2)

Timer	Periodic Temporal clustering Sequential clustering
State dependent control	Control Control clustering
Coordinator	Coordinator Sequential clustering
Business logic	Asynchronous business logic Periodic business logic Any clustering criterion
Algorithm	Asynchronous algorithm Periodic algorithm Non-time-critical Any clustering criterion

DSIE (UPCT)

182

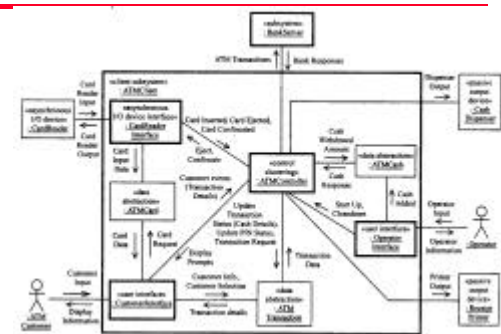
D. Colaboración Concurrente Preliminar

- Tras estructurar el sistema en tareas concurrentes se dibuja el D. de Colaboración Concurrente inicial incluyendo todas las tareas del sistema.
- Es preliminar porque los interfaces son simples tal como se incluían en el análisis.

DSIE (UPCT)

183

D. Colaboración Concurrente Preliminar



DSIE (UPCT)

184

Comunicación entre Tareas y Sincronización

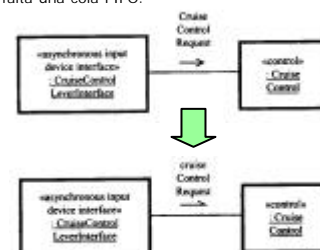
- El siguiente paso es definir las interfaces entre las tareas:
 - Comunicación por mensajes.
 - Asíncrona.
 - Síncrona con respuesta.
 - Síncrona sin respuesta.
 - Sincronización de eventos.
 - Interacción mediante objetos ocultadores de información.

DSIE (UPCT)

185

Comunicación entre Tareas y Sincronización

- Comunicación Asíncrona:
 - Hace falta una cola FIFO.



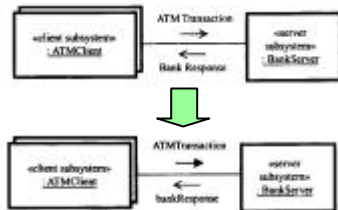
DSIE (UPCT)

186

Comunicación entre Tareas y Sincronización

Comunicación Síncrona con respuesta:

- El productor manda un mensaje al consumidor y espera respuesta. Cuando el mensaje llega se acepta el mensaje, se procesa, se genera la respuesta y se manda. El consumidor se suspende si no hay mensaje disponible.

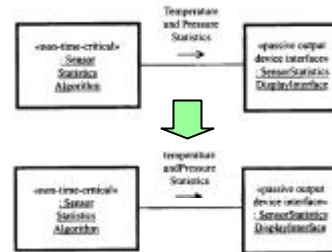


DSIE (UPCT)

187

Comunicación entre Tareas y Sincronización

Comunicación Síncrona sin respuesta:



DSIE (UPCT)

188

Comunicación entre Tareas y Sincronización

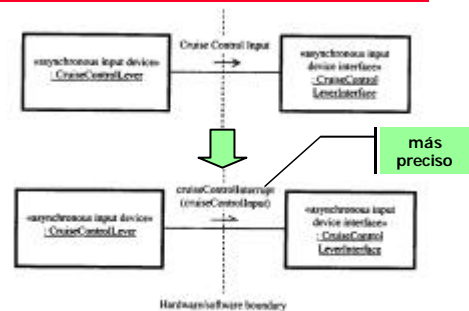
Sincronización de eventos:

- Por evento externo
- Por evento interno
- Por evento timer
- Un **evento externo** viene de una entidad externa, típicamente una interrupción desde un dispositivo I/O externo.
- Un **evento interno** representa sincronización interna entre dos tareas. **No hay transferencia de datos.** La destino espera al evento y queda suspendida. No se suspende si previamente la tarea origen señaló el evento.
- Un **evento timer** representa una activación periódica de una tarea.
- Los eventos se denotan en UML mediante mensajes asíncronos.

DSIE (UPCT)

189

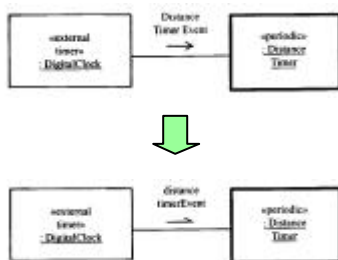
Ejemplo de evento externo



DSIE (UPCT)

190

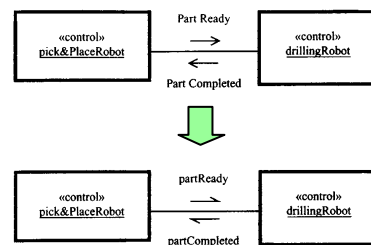
Ejemplo de evento timer



DSIE (UPCT)

191

Ejemplo de evento de sincronización interno



DSIE (UPCT)

192

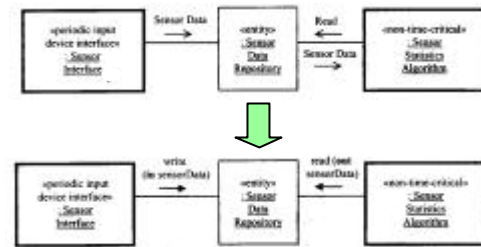
Comunicación entre Tareas y Sincronización

- Interacción mediante objetos ocultadores de información.
 - ✦ La interacción con el objeto pasivo es mediante una llamada a operación.
 - ✦ En UML es la misma notación que la usada para la comunicación síncrona pero es una llamada a un objeto pasivo.

DSIE (UPCT)

193

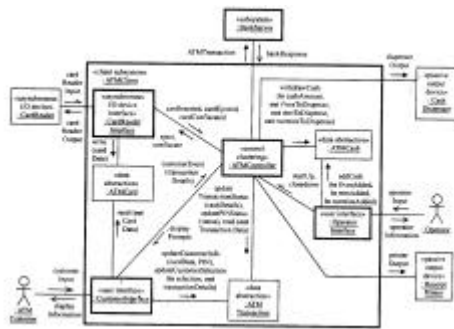
Ejemplo de Interacción a través de objeto pasivo



DSIE (UPCT)

194

D. Colaboración Concurrente Revisado



DSIE (UPCT)

195

Especificación de Tareas

- Mediante la Especificación del Comportamiento de Tareas se describe:
 - ✦ El interfaz
 - ✦ La estructura
 - ✦ Los aspectos temporales
 - ✦ La prioridad
 - ✦ La lógica de la secuencia de eventos
 - ✦ Los errores detectados.
- Es decir, ¿cómo se conectan entre sí las tareas?.

DSIE (UPCT)

196

Especificación de Tareas (1/2)

- Interfaz de la Tarea**
 - ✦ Mensajes de entrada y salida. Asíncrono, síncrono con o sin respuesta, nombre y parámetros.
 - ✦ Eventos señalados (entrada y salida). Nombre y tipo (externo, interno y timer).
 - ✦ Entradas y salidas externas. I/O al entorno.
 - ✦ Objetos pasivos referenciados.
- Información de la estructura de la Tarea**
 - ✦ Criterio de estructuración usado para diseñar la tarea.
 - ✦ Objetos que han dado lugar a la tarea.

DSIE (UPCT)

197

Especificación de Tareas (2/2)

- Características Temporales**
 - ✦ Frecuencia de activación. Si periódica determinar el periodo. Si asíncrona estimar máximo y media de la activación por eventos externos.
 - ✦ Tiempo estimado de ejecución. Todas las alternativas posibles.
- Prioridad de la Tarea**
- Errores detectados en ejecución por la Tarea**
- Lógica de secuencia de eventos**
 - ✦ Cómo responde la tarea a cada entrada recibida.

DSIE (UPCT)

198

Ej. de Especificación de Tareas

1. INTERFAZ DE LA TAREA

ENTRADAS

Síncrona con respuesta:

Mensajes:

validatePIN

Entradas: cardID, PIN

Respuestas: PINValidationResponse

withdraw

Entradas: cardID, account#, amount

Respuestas: withdrawalResponse

query

Entradas: cardID, account#

Respuestas: queryResponse

DSIE (UPCT)

199

Ej. de Especificación de Tareas

transfer

Entradas: cardID, fromAccount#, toAccount#, amount

Respuestas: transferResponse

SALIDAS:

las especificadas en los mensajes anteriores

2. ESTRUCTURA

Criterio: Agrupación secuencial

Objetos origen: Bank Transaction Server,

PIN Validation Transaction Manager

Withdrawal Transaction Manager

Query Transaction Manager

Transfer Transaction Manager

Checking Account

Savings Account

Debit Card

Card Account

Transaction Log.

DSIE (UPCT)

200

Ej. de Especificación de Tareas

CARACTERÍSTICAS TEMPORALES

Activación: Asíncrona. Peor caso 100 msec. Media entre llegadas > 1 seg.

Tiempo Ejecución: 10msecs por mensaje

PRIORIDAD: Elevada

ERRORES DETECTADOS: Mensaje no reconocido

LÓGICA SECUENCIA DE EVENTOS:

La lógica se define más adelante en el diseño detallado del Sw.

DSIE (UPCT)

201

Conclusión

- *Los criterios de estructuración asisten en el paso del modelo de análisis a la arquitectura concurrente del sistema.*
- *Cada tarea ejecuta una serie de operaciones.*
- *Los objetos que proporcionan estas operaciones no se definen hasta la fase de diseño.*

DSIE (UPCT)

202

Diseño de Clases

DSIE (UPCT)

203

Introducción

- Abordaremos el diseño de las clases ocultadoras de información de los que se instanciarán objetos pasivos.
- Determinaremos las operaciones de las clases a partir del modelo dinámico.

DSIE (UPCT)

204

Diseño de Clases Ocultadoras de Información

- Las clases del modelo de análisis se clasificaban en entidades, interfaz, de control y de la lógica de la aplicación.
- Surgen aquí las **clases decisión del software**.
- Clases Entidad «Entity»:**
 - Vienen del análisis y encapsulan datos.
 - «data abstraction classes» encapsulan estruct. de datos.
 - «wrapper classes» proveen interfaz a sist. archivos.
 - «database wrapper classes» proveen interfaz a BDs.
 - wrappers también proveen interfaz a sistemas legacy.

DSIE (UPCT)

205

Diseño de Clases Ocultadoras de Información

- Clases Interfaz:**
 - Clases interfaz a dispositivo. Dispositivos Hw I/O.
 - Clases interfaz a usuarios.
 - Clases interfaz a sistemas. Externos o subsistemas.
- Clases Control:**
 - Clases coordinador.
 - Clases dependientes del estado.
 - Clases timer.
- Clases Lógica de la Aplicación.**
 - Clases de la lógica de negocios.
 - Clases algoritmos.
- Clases decisión de Software:**
 - Esconden decisiones de diseño que pueden cambiar.
 - Surgen en el diseño.

DSIE (UPCT)

206

Diseño de Operaciones desde el D. de Colaboración

- Las operaciones se determinan con facilidad desde el diagrama de colaboración.
- Cuando dos objetos interactúan uno provee una operación que es usada por el otro.
- El diagrama de clases no es suficiente para determinar quién de los dos objetos relacionados es quien invoca al otro.
- El nombre del mensaje será el de la operación.
- Idem con los parámetros.

DSIE (UPCT)

207

Diseño de Operaciones desde el D. de Colaboración

- Lo que suele ocurrir es que el nombre del mensaje del d. de colaboración no es preciso.
- Puede ser un nombre (los datos pasados) o un verbo (la operación a ejecutar).
- El nombre de la operación debe reflejar el servicio provisto.
- Se debe saber además si hay parámetros y si son de entrada o salida.

DSIE (UPCT)

208

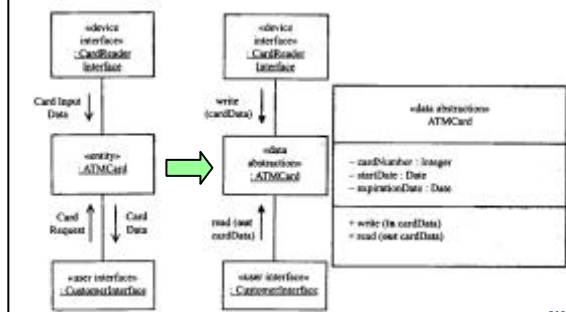
Diseño de Operaciones desde el D. de Colaboración

- Algunos mensajes son respuesta a otros previos.
- Todas las invocaciones de operaciones se muestran como mensajes síncronos.
- Los mensajes respuesta se incluyen como parámetros de respuesta de la operación.
- Conforme se determinan operaciones se va completando el modelo estático.

DSIE (UPCT)

209

Diseño de Operaciones desde el D. de Colaboración



DSIE (UPCT)

210

Diseño de Operaciones desde los StateCharts

- Un statechart incluye acciones y actividades iniciadas como resultado de la transición de estados.
- Las acciones suelen ser operaciones de clases o bien de tareas concurrentes.
- Determinadas acciones pueden estar presentes sólo el D. de Colaboración Consolidado.

DSIE (UPCT)

211

Diseño de Operaciones desde el modelo estático

- Suele ser habitual determinar las operaciones desde el modelo estático sobre todo para las clases entidad.
- Operaciones estandar son `create`, `read`, `update`, `delete`,...

DSIE (UPCT)

212

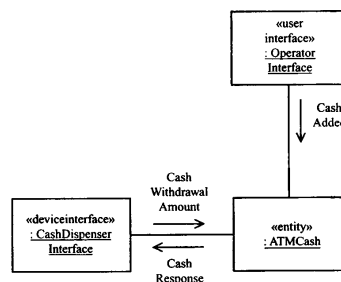
Clases Abstracción de Datos

- Cada clase entidad del modelo de datos se diseña como una **clase abstracción de datos**.
- Encapsulan una estructura de datos y proveen operaciones para acceder a dichos datos.
- Los atributos de estas clases suelen estar disponibles desde el modelo estático del dominio del problema.
- Las operaciones se determinan a partir de los servicios requeridos por los objetos clientes que usan la abstracción con vistas a acceder a la estructura de datos encapsulada.

DSIE (UPCT)

213

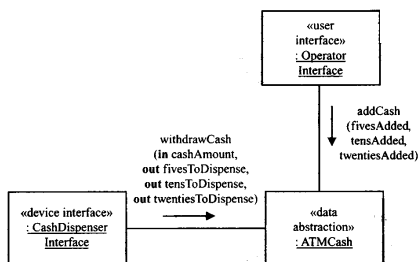
Ejemplo de Clase Abstracción de Datos



DSIE (UPCT)

214

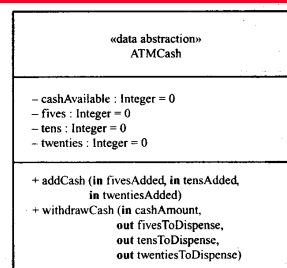
Ejemplo de Clase Abstracción de Datos



DSIE (UPCT)

215

Ejemplo de Clase Abstracción de Datos



Invariante
 $\text{cashAvailable} = 5 * \text{fives} + 10 * \text{tens} + 20 * \text{twenties}$

DSIE (UPCT)

216

Clases Interfaz de Dispositivo

- Una clase interfaz de dispositivo proporciona un interfaz virtual que oculta el interfaz real de un interfaz de dispositivo I/O.
- Se oculta la decisión de diseño de cómo conectarse al dispositivo en cuestión.
- Los usuarios acceden al dispositivo a través del interfaz virtual.
- Cambios en el dispositivo afectan sólo a la clase interfaz.

DSIE (UPCT)

217

Operaciones de las Clases Interfaz de Dispositivo

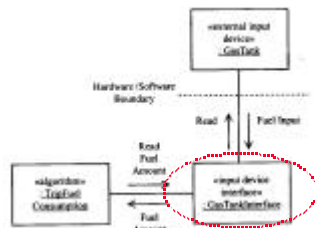
- Las operaciones proporcionan las operaciones que leen y escriben en el dispositivo real.
- Siempre hay una operación `initialize`.
- Las otras operaciones dependen del dispositivo.

DSIE (UPCT)

218

Clases Interfaz de Dispositivos de Entrada

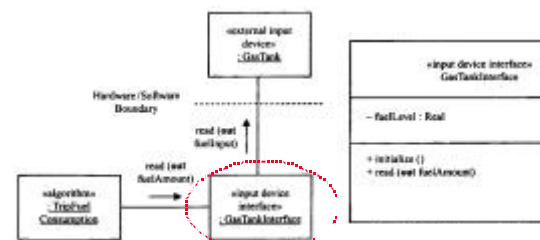
- Necesita una operación de lectura `read`.
- La implementación es totalmente dependiente del dispositivo y oculta a los usuarios del mismo.



DSIE (UPCT)

219

Clases Interfaz de Dispositivos de Entrada



DSIE (UPCT)

220

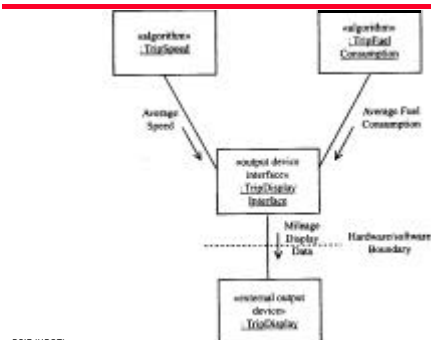
Clases Interfaz de Dispositivos de Salida

- También tiene operación `initialize`.
- Debe tener una operación `output`.

DSIE (UPCT)

221

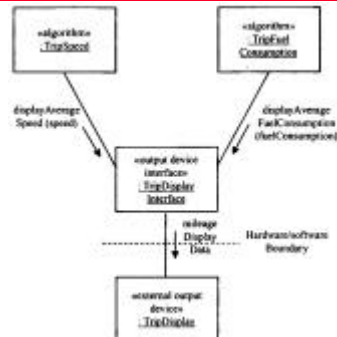
Clases Interfaz de Dispositivos de Salida



DSIE (UPCT)

222

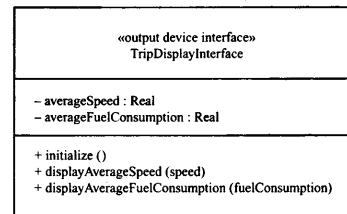
Clases Interfaz de Dispositivos de Salida



DSIE (UPCT)

223

Clases Interfaz de Dispositivos de Salida



DSIE (UPCT)

224

Clases Dependientes del Estado

- Encapsulan la información de un statechart.
- El statechart se traduce a una tabla de transición de estado.
- Estas clases encapsulan la información del statechart y mantienen el valor actual del estado.
- Proveen las operaciones que acceden a la tabla y que cambian el estado.
- Una forma es tener una operación para cada evento entrante.

DSIE (UPCT)

225

Clases Dependientes del Estado

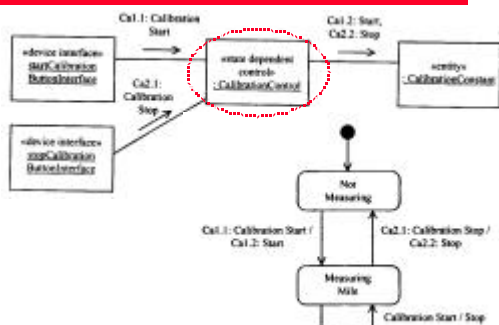
- Otra forma es tener una operación `processEvent` que devuelva las acciones por procesar dado un evento de entrada:

```
processEvent (in event, out action)
```

DSIE (UPCT)

226

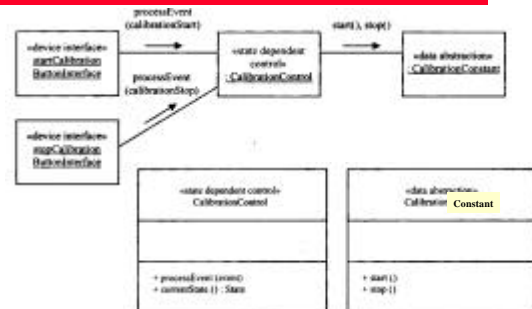
Clases Dependientes del Estado



DSIE (UPCT)

227

Clases Dependientes del Estado



DSIE (UPCT)

228

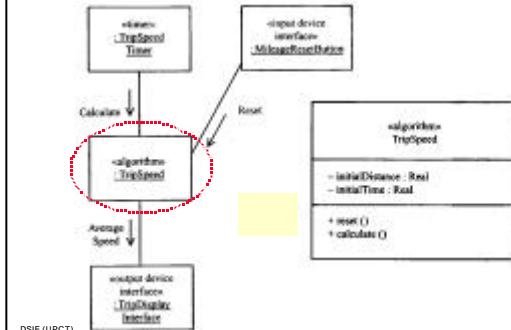
Clases ocultadoras de Algoritmos

- Encapsulan algoritmos del dominio del problema.
- Típicas de real-time y de aplicaciones ingenieriles.
- Se encapsulan tanto datos como el algoritmo en sí.

DSIE (UPCT)

229

Clases ocultadoras de Algoritmos



DSIE (UPCT)

230

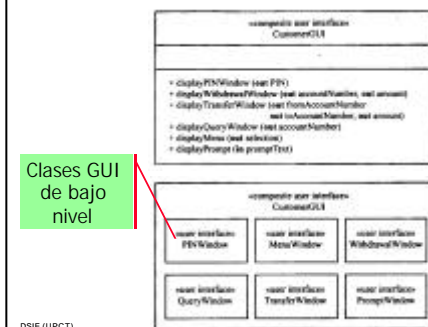
Clases Interfaz de Usuario

- Oculto los detalles relativos al interfaz con los usuarios.
- Puede ser simple línea de comandos o GUIs.
- El interfaz de usuario puede formar parte del modelo de análisis.

DSIE (UPCT)

231

Clases Interfaz de Usuario



DSIE (UPCT)

232

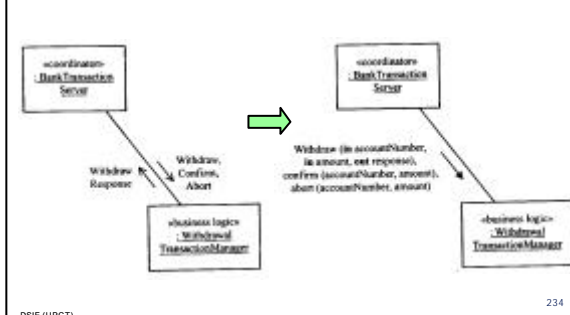
Clases de la Lógica de Negocios

- Representan toma de decisiones específicos del negocio de la aplicación.
- Encapsulan reglas que pueden cambiar independientemente de otros aspectos.
- Estas clases acceden a varias clases entidad.

DSIE (UPCT)

233

Clases de la Lógica de Negocios



DSIE (UPCT)

234

Database Wrapper Classes

- En análisis una entidad encapsula datos.
- En diseño se decide cómo se manejan dichos datos.
- Si se manejan en la entidad se llaman data abstraction classes.
- Si se manejan en una base de datos se llaman **database wrapper classes**.
- Estas clases ocultan cómo se acceden a los datos.

DSIE (UPCT)

235

Database Wrapper Classes

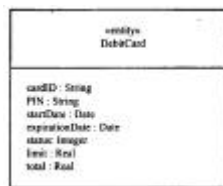
- Proveen un interfaz orientado a objetos a la clase.
- Cuando la BD es Relacional las entidades del modelo estático se pasan a la BD.
- El paso a un modelo relacional no es trivial ni presenta una solución única.

DSIE (UPCT)

236

Database Wrapper Classes

- La clase database wrapper oculta cómo acceder a los datos, esto es, oculta las sentencias SQL y los detalles de las relaciones y vistas.



DSIE (UPCT)

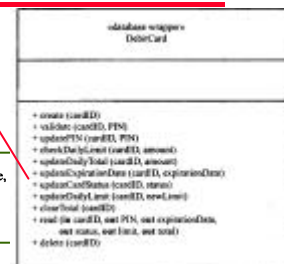
237

Database Wrapper Classes

Encapsulan el SQL para acceder a la relación

Relation in relational database :
DebitCard (cardID, PIN, startDate, expirationDate, status, limit, total, *customerSSN*)

(underline = primary key, italic = *foreign key*)



DSIE (UPCT)

238

Clases Decisión Software

- Ocultan decisiones de diseño hechas por el diseñador que pueden cambiar.
- Suelen desarrollarse tarde en el proceso.
- No se deducen del dominio del problema (análisis).
- Por ejemplo: pilas, colas, tablas de datos, conectores, ...

DSIE (UPCT)

239

Especificación del Interfaz de Clases

- Definen el interfaz de la clase ocultadora de información incluyendo las operaciones:
 - Información oculta (estructura de datos, dispositivo,...)
 - Criterio de estructuración usado.
 - Cambios que se anticipan.
 - Superclase (si existe).
 - Operaciones heredadas (si existen).
 - Operaciones de la clase:
 - Función ejecutada.
 - Precondición.
 - Postcondición.
 - Invariante.
 - Parámetros de entrada y salida.
 - Operaciones usadas de otras clases.

DSIE (UPCT)

240

Diseño Detallado de Software

DSIE (UPCT)

241

Introducción

- Tras diseñar las clases en tareas y las clases ocultadoras de información es preciso diseñar los aspectos internos a las tareas como composición, sincronización y comunicación.
- Se desarrolla el **D. de Colaboración Concurrente y Detallado** de cada subsistema.

DSIE (UPCT)

242

Diseño de Tareas Compuestas

- Las tareas estructuradas por los criterios de **agrupación e inversión** se diseñan como **clases activas compuestas** que anidan a clases pasivas.

DSIE (UPCT)

243

Relación entre Tareas y Clases

- El objeto activo (tarea) es activado por un evento externo, interno o timer y llama a una operación provista por un objeto pasivo.
- El objeto pasivo puede estar dentro de la tarea o fuera de ella.
- Cuando las operaciones se usan de forma exclusiva por la tarea entonces puede anidarse en ella.

DSIE (UPCT)

244

Relación entre Tareas y Clases

- Si es accedida por más de una tarea entonces las operaciones de la clase deben sincronizar el acceso a los datos encapsulados.
- Es necesario definir el contexto en el que la clase va a usarse.
- Puede no ser deseable (reusabilidad) el anidar las clases físicamente cuando son accedidas por una sola clase pero implica especificar claramente los aspectos de sincronización.

DSIE (UPCT)

245

División de Responsabilidades

- A veces es útil dividir responsabilidades entre una tarea y una clase anidada.
- Control, secuenciación y responsabilidades de comunicación a la tarea.
- Aspectos estructurales a la clase anidada.

DSIE (UPCT)

246

Ej. de División de Responsabilidades

- Input/ output device: puede usarse una tarea de interfaz a dispositivo (asíncrono o periódico) con un objeto anidado de interfaz a dispositivo.
- El objeto trata el cómo leer y escribir del dispositivo.
- La tarea trata cuando y cómo la tarea se activa y se comunica con otros objetos (pasivos o activos).

DSIE (UPCT)

247

Ej. de División de Responsabilidades

- La tarea se activa mediante evento externo o timer y llama a una operación provista por el objeto pasivo para leer.
- Si el objeto interfaz no es accedido por otra tarea no hace falta aspectos de sincronización.

DSIE (UPCT)

248

Otro Ej. de División de Responsabilidades

- División entre una tarea de control y el objeto anidado dependiente del estado.
- La tarea de control recibe los mensajes y lo encamina al objeto anidado.
- El objeto anidado devuelve la acción que se debe ejecutar.
- La tarea inicia la acción mandando un mensaje o invocando una operación de otro objeto.

DSIE (UPCT)

249

Tareas Compuestas

- Son tareas que encapsulan los objetos anidados que contienen.
- Se utiliza el D. de Colaboración Concurrente detallado.
- Cada tarea compuesta contiene un objeto coordinador que recibe mensajes y lo traduce a invocaciones de operaciones.

DSIE (UPCT)

250

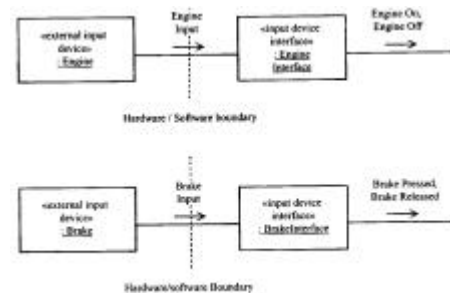
Tareas Agrupadas Temporalmente y Obj. Interfaz de Dispositivo

- Caso: una tarea estructurada como compuesta que contiene tres objetos (uno coordinador y dos interfaz de dispositivo).
- Ambos dispositivos se muestrean periódicamente y con la misma frecuencia.
- Objetivo: separar cómo se accede a los dispositivos de cuándo.
- Mayor flexibilidad y reusabilidad.

DSIE (UPCT)

251

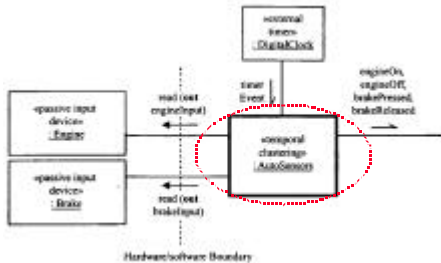
Tareas Agrupadas Temporalmente y Obj. Interfaz de Dispositivo



DSIE (UPCT)

252

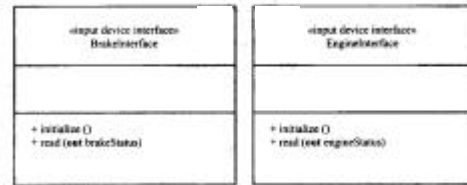
Tareas Agrupadas Temporalmente y Obj. Interfaz de Dispositivo



DSIE (UPCT)

253

Tareas Agrupadas Temporalmente y Obj. Interfaz de Dispositivo

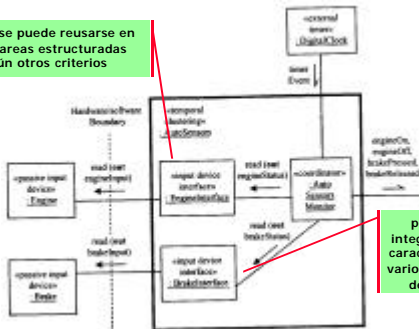


DSIE (UPCT)

254

Tareas Agrupadas Temporalmente y Obj. Interfaz de Dispositivo

Esta clase puede reusarse en otras tareas estructuradas según otros criterios



DSIE (UPCT)

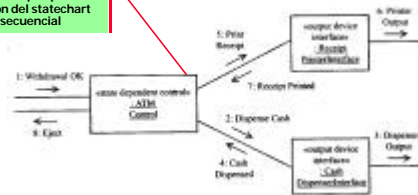
255

pueden integrarse las característ. de varios sensores de freno

Tareas agrupadas por Control y Objetos Ocultadores de Información

- Caso: una tarea control activada asincrónicamente que llama a operaciones provistas por uno o más objetos pasivos.

Se estructura como tarea de control porque la ejecución del statechart es secuencial

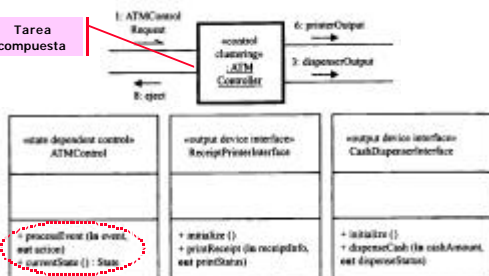


DSIE (UPCT)

256

Tareas agrupadas por Control y Objetos Ocultadores de Información

Tarea compuesta

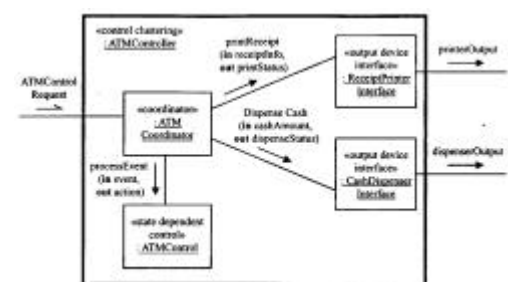


estructura + contenido de la tabla de transición estado

DSIE (UPCT)

257

Tareas agrupadas por Control y Objetos Ocultadores de Información



DSIE (UPCT)

258

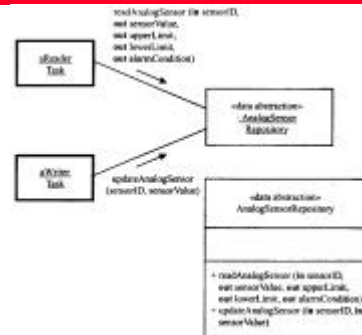
Sincronización del Acceso a Clases

- Si una clase es accedida por más de una tarea las operaciones de la clase deben sincronizar el acceso a los datos que encapsula.
- ¿Qué algoritmo de sincronización se usa?
- Separación del *qué* frente al *cómo*.
- Cambios posibles:
 - ✦ Cambios a la estructura interna de la estructura de datos (pila, cola,...).
 - ✦ Cambios relativos a la sincronización en el acceso (de exclusión mutua a lectores/escritores).
- Estos cambios no afectan al *qué*.

DSIE (UPCT)

259

Ejemplo de Sincronización del Acceso a Clases



DSIE (UPCT)

260

Usando Exclusión Mutua

- TEXTP 16.2.3

DSIE (UPCT)

261

Diseño de Conectores para Comunicación entre Tareas

- Los **Conectores** encapsulan los detalles de la comunicación entre tareas tales como la comunicación mediante mensajes.
- Algunos lenguajes de programación concurrente (Ada, **Java**) proveen mecanismos para comunicación inter-tareas y sincronización.
- Ninguno de estos lenguajes proporciona comunicación asíncrona de manera directa.

DSIE (UPCT)

262

Diseño de Conectores para Comunicación entre Tareas

- La comunicación asíncrona se consigue utilizar una clase ocultadora de información que haga de cola de mensajes.
- Este tipo de clases se llaman **conectores**.
- Se usa un **monitor** según el tipo de comunicación.
- Cada monitor es un conector.

DSIE (UPCT)

263

Diseño de un Conector Cola de Mensajes

- Utilizado para encapsular el mecanismo de **comunicación asíncrona**.
- Monitor que encapsula una cola.
- Provee operaciones sincronizadas para enviar mensajes y recibir mensajes.
- El productor se suspende si la cola está llena.
- El productor se reactiva cuando queda un hueco libre.

DSIE (UPCT)

264

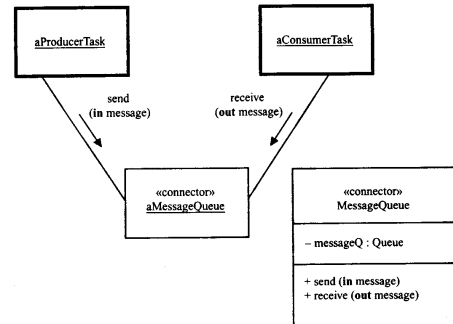
Diseño de un Conector Cola de Mensajes

- Tras añadir el mensaje en la cola el productor continúa su ejecución enviando otros mensajes.
- El consumidor se suspende si no tiene mensajes en la cola.
- Se asumen varios productores para un consumidor.

DSIE (UPCT)

265

Diseño de un Conector Cola de Mensajes



DSIE (UI)

266

Diseño de un Conector Cola de Mensajes

```

monitor MessageQueue
private maxCount : Integer :=0;
public send (in message)
    while messageCount = maxCount do wait;
    place message in buffer;
    Increment messageCount;
    if messageCount = 1 then signal;
end send;

public receive (out message)
    while messageCount = 0 do wait;
    remove message from buffer;
    decrement messageCount;
    if messageCount = maxCount - 1 then signal;
end receive;
end MessageQueue;
    
```

DSIE (UPCT)

267

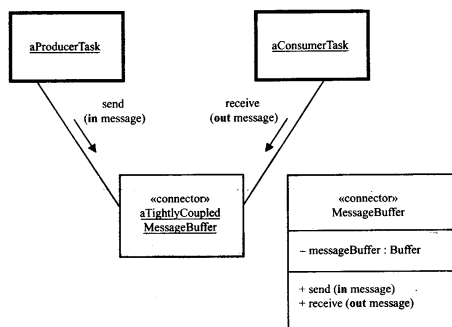
Diseño de un Conector Buffer de Mensajes

- Usado para encapsular el mecanismo de comunicación síncrona sin respuesta.
- El productor se suspende si el buffer está lleno.
- El productor se suspende hasta que es atendido el mensaje dejado en el buffer.
- El consumidor se suspende si el buffer está vacío.
- Un productor y un consumidor.

DSIE (UPCT)

268

Diseño de un Conector Buffer de Mensajes



DSIE

269

Diseño de un Conector Buffer de Mensajes

```

monitor MessageBuffer
private messageBufferFull : Boolean :=false;
public send (in message)
    place message in buffer;
    messageBufferFull := true;
    signal;
    while messageBufferFull = true do wait;
end send;

public receive (out message)
    while messageBufferFull =false do wait;
    remove message from buffer;
    messageBufferFull := false;
    signal;
end receive;
end MessageBuffer;
    
```

DSIE (UPCT)

270

Diseño de un Buffer de Mensajes y Conector de Respuesta

- Usado para encapsular el mecanismo de **comunicación síncrona con respuesta**.
- El conector es un monitor que encapsula un buffer simple de mensaje y otro de respuesta simple.
- Se proveen operaciones sincronizadas para **enviar**, **recibir** y **enviar respuesta**.
- El productor envía, el consumidor recibe y manda la respuesta.

DSIE (UPCT)

271

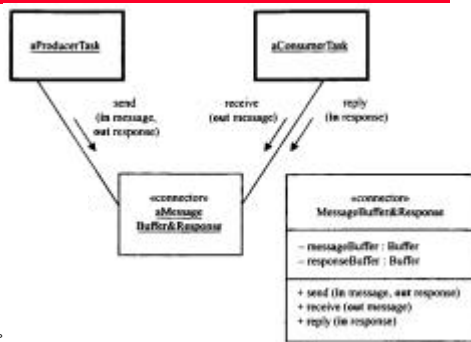
Diseño de un Buffer de Mensajes y Conector de Respuesta

- Tras colocar el mensaje en el buffer el productor se suspende hasta recibir la respuesta.
- El consumidor se suspende si el buffer está vacío.
- Un productor y un consumidor.

DSIE (UPCT)

272

Diseño de un Buffer de Mensajes y Conector de Respuesta



DSIE (UP)

273

Diseño de un Buffer de Mensajes y Conector de Respuesta

```
monitor MessageBuffer&Response
private messageBufferFull : Boolean :=false;
private responseBufferFull : Boolean :=false;
public send (in message, out response)
  place message in buffer;
  messageBufferFull := true;
  signal;
  while responseBufferFull = false do wait;
  remove response from response buffer;
  responseBufferFull := false;
end send;

public receive (out message)
  while messageBufferFull = false do wait;
  remove message from buffer;
  messageBufferFull := false;
end receive;
```

DSIE (UPCT)

274

Diseño de un Buffer de Mensajes y Conector de Respuesta

```
public reply (in response)
  Place response in response buffer;
  responseBufferFull := true;
  signal;
end reply;
end MessageBuffer&Response;
```

DSIE (UPCT)

275

Diseño de Tareas Cooperantes usando Conectores

- Consideramos un grupo de tareas cooperantes que se comunican mediante un objeto conector.
- Una tarea envía un mensaje y otra lo recibe a través del conector.

DSIE (UPCT)

276

Conclusión

- Se han diseñado los detalles internos de aquellas tareas formadas por otros objetos.
- Se han tratado aspectos de sincronización.
- Se han diseñado conectores que encapsulan la comunicación entre tareas.
- Se han definido los detalles internos de cada tarea mediante la lógica de secuenciación de eventos.

DSIE (UPCT)

283

Análisis de Productividad de Diseños Sw Concurrentes y de Tiempo Real

DSIE (UPCT)

284

Introducción

- El estudio de la productividad de diseños Sw es particularmente importante en sistemas concurrentes y de tiempo real.
- Un fallo puede dar lugar a una catástrofe.
- La detección temprana de problemas de productividad permite el estudio de alternativas de diseño.

DSIE (UPCT)

285

Introducción

- COMET considera dos enfoques:
 - ✦ La Teoría de Planificación de Tiempo Real.
 - ✦ El análisis de Secuencia de Eventos.

DSIE (UPCT)

286

TPTR

- La Teoría de Planificación de Tiempo Real estudia los aspectos de planificación de tareas concurrentes a partir de prioridades con vistas a satisfacer restricciones fuertes de tiempo real.
 - ¿Un grupo de tareas cuya utilización de CPU se conoce cumplirán sus deadlines?
- Teorema de Acotamiento de Utilización.
- Teorema de Compleción de Tiempos.
- Tareas Aperiódicas.
- Inversión de Prioridades.
- Teorema de Acotamiento de Utilización Generalizado.

DSIE (UPCT)

287

Reflexión Final

DSIE (UPCT)

288

Conclusiones Globales

- UML es una notación, no una metodología.
- Distintas propuestas usan UML con distintas metodologías y procesos.
- Los sistemas concurrentes y de tiempo real poseen peculiaridades específicas.
- COMET usa UML con estereotipos para representar los distintos elementos de los sistemas.

DSIE (UPCT)

289

Conclusiones Globales

- COMET se centra en la fase de Diseño.
- Aporta heurísticos para el particionamiento y clasificación de los distintos elementos que forman un sistema.
- Desarrolla 4 casos de estudio completos con características distintas.
- ¿ Profile UML – Rational Rose ?

DSIE (UPCT)

290