
Diseño de Aplicaciones Concurrentes, Distribuidas y de Tiempo Real usando UML

Dr. Pedro Sánchez Palma
pedro.sanchez@upct.es
Universidad Politécnica de Cartagena
Curso 2001/2002

DSIE (UPCT)

1

PARTE I

UML, Conceptos de Diseño,
Tecnologías, Ciclos de Vida y
Métodos

DSIE (UPCT)

2

Contenido Parte I

- UML, Conceptos de Diseño, Tecnologías, Ciclos de Vida y Métodos:
 - ✕ Introducción : Metodología COMET
 - ✕ Tour por UML
 - ✕ Aspectos de comunicación
 - ✕ Tecnología para Sistemas Concurrentes y Distribuidos
 - ✕ El Ciclo de Desarrollo de Software
 - ✕ Detalle de COMET

DSIE (UPCT)

3

Introducción: Metodología COMET

DSIE (UPCT)

4

COMET

- Concurrent **O**bject **M**odeling and architectural design **mET**hod.
- Dr. Hassan Gomaa. George Mason University.
- "Designing Concurrent Distributed and Real-Time Applications with UML". Addison-Wesley (1998).

DSIE (UPCT)

5

COMET

- Utiliza la notación **UML** para describir los sistemas.
- Se basa en conceptos como:
 - ✕ Ocultación de Información
 - ✕ Clases
 - ✕ Herencia
 - ✕ Tareas Concurrentes
 - ✕ ...

DSIE (UPCT)

6

COMET

- Estrategia de diseño.
- **Sistema** = conjunto de **objetos** pasivos y activos + **interfaces** entre ellos.
- Provee **heurísticos** y criterios para descomponer un sistema en subsistemas + objetos que los constituyen.

DSIE (UPCT)

7

Aplicaciones Concurrentes?

- *La naturaleza es inherentemente concurrente...*
- Una aplicación concurrente posee varias actividades funcionando en **paralelo**.
- Típico de sistemas de **Tiempo-Real** y **Distribuidos**.

DSIE (UPCT)

8

Problemas secuenciales y concurrentes

- **Secuenciales:**
 - ✦ Las actividades tienen lugar en secuencia estricta.
 - ✦ P.e. Las nóminas de una empresa se procesan secuencialmente para cada empleado.
- **Concurrentes:**
 - ✦ Las actividades ocurren en paralelo.
 - ✦ P.e. Varios usuarios que interactúan con un sistema simultáneamente.
 - ✦ P.e. Un sistema de control aéreo que monitoriza varios aviones simultáneamente.

DSIE (UPCT)

9

Problemas secuenciales y concurrentes

- Muchas aplicaciones son intrínsecamente concurrentes.
- Poseen varias actividades funcionando en paralelo.
- Llevar estas actividades concurrentes a secuenciales supone un diseño muy complejo.

DSIE (UPCT)

10

Problemas secuenciales y concurrentes

- Es más sencillo diseñar actividades **separadas**.
- Cada tarea separada funciona concurrentemente con el resto aunque eventualmente **sincronicen** sus operaciones o **intercambien** información.

DSIE (UPCT)

11

Aplicaciones secuenciales

- Una aplicación **secuencial** está formada por **objetos pasivos** y tiene **un solo hilo** de control.
- El único hilo de control se transfiere de un objeto a otro cuando es invocado.

DSIE (UPCT)

12

Aplicaciones concurrentes

- Una aplicación **concurrente** está formada por **objetos activos** donde cada uno posee **su propio hilo** de control.
- Existe la comunicación **asíncrona** entre objetos.

DSIE (UPCT)

13

Tareas (procesos) concurrentes

- Una **aplicación concurrente** está formada por varias tareas concurrentes que se ejecutan en paralelo e interaccionan entre sí.
- Habitual en sistemas de Tiempo-Real y Distribuidos.

DSIE (UPCT)

14

Sistemas de Tiempo-Real (STR)

- Un **STR** es un sistema concurrente con restricciones temporales de ejecución.
- Aplicaciones industriales, militares, comerciales, ...
- El término "tiempo-real" se refiere a la totalidad del sistema (S.O., I/O, aplicación,...)

DSIE (UPCT)

15

Sistemas de Tiempo-Real

- STR Hard**: las restricciones temporales deben satisfacerse para evitar catástrofes.
- STR Soft**: las restricciones temporales son deseables pero no vitales.

DSIE (UPCT)

16

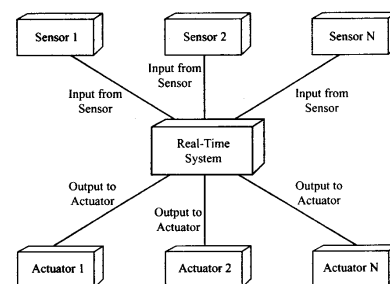
Características de los STRs

- Sistemas empujados.
- Interacción con el entorno (sensores y actuadores).
- Restricciones temporales.
- Control en tiempo real.
- Sistemas reactivos.

DSIE (UPCT)

17

STR

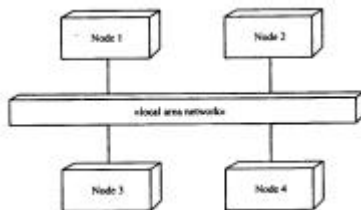


DSIE (UPCT)

18

Sistemas distribuidos (SD)

- Un SD es una aplicación concurrente que se ejecuta en varios **nodos geográficamente distribuidos y conectados** entre sí.



DSIE (UPCT)

19

Sistemas distribuidos (SD)

- Ventajas del procesamiento distribuido:
 - ✦ La disponibilidad del sistema es mayor.
 - ✦ Configuración más flexible.
 - ✦ Control y manejo localizado por nodo.
 - ✦ Expansión incremental de sistemas.
 - ✦ Coste reducido (microcomputadores).
 - ✦ Carga del sistema repartida.
 - ✦ Tiempo de respuesta optimizado.

DSIE (UPCT)

20

Tour por UML
[Curso UML reducido.ppt](#)

DSIE (UPCT)

21

Aspectos de Comunicación

DSIE (UPCT)

22

Procesamiento concurrente

- Cada tarea contiene un hilo de control: no consideramos concurrencia intra-tarea.
- La concurrencia viene por la ejecución concurrente (asíncrona) de todas las tareas.
- De vez en cuando sincronizan y se comunican entre sí.

DSIE (UPCT)

23

Procesamiento concurrente

- El procesamiento concurrente tiene sus orígenes a principios de los años 70.
- Libros, Sw, artículos, ...
- Sistemas operativos, algoritmos para comunicación y sincronización, ...
- Ejemplos: lectores/escritores, filósofos pensantes, ...

DSIE (UPCT)

24

Procesamiento concurrente

- **Ventajas:**
 - » Refleja de manera natural a muchos sistemas.
 - » Sistemas más fáciles de entender, manejar y construir.
 - » Puede reducirse mejor la carga de los sistemas (multiprocesadores...)
 - » Las tareas críticas con mayor prioridad: planificación más flexible.
 - » Análisis preliminar de la productividad del sistema (modelado con Redes de Petri, planificación Real-Time, ...).
- **Pero...**
 - » Demasiadas tareas en un sistema puede llevar a un incremento innecesario de la complejidad y la carga del sistema.
 - » ¿ Demasiada comunicación e interacción ?

DSIE (UPCT)

25

Procesos ligeros y pesados

- **Procesos pesados:**
 - » Espacio propio de direcciones.
 - » Cambio de contexto muy pesado.
 - » Independencia entre procesos.
- **Procesos ligeros (threads):**
 - » Comparten un espacio de direcciones.
 - » Cambio de contexto ligero.
 - » Crash en un => crash en todos.
- Un proceso pesado posee varios threads.
- Nos referiremos a **tarea** como thread o proceso.

DSIE (UPCT)

26

Cooperación entre tareas concurrentes

- El diseño de sistemas concurrentes necesita dar solución a problemas que no existen en los secuenciales:
 - » Problema de **exclusión mutua**: una tarea necesita **acceso exclusivo a determinados recursos**.
 - » Problema de la **sincronización de tareas**: dos tareas **necesitan sincronizar sus operaciones entre sí**.
 - » Problema **productor/consumidor**: una tarea se **comunica con otra para pasar información**.

DSIE (UPCT)

27

Problema de exclusión mutua

- Surge cuando un recurso necesita ser accedido de forma exclusiva por una tarea a la vez.
- Si hay concurrencia entonces más de una tarea puede necesitar acceder simultáneamente al mismo recurso.
- Hace falta sincronizar el acceso para garantizar la exclusividad.

DSIE (UPCT)

28

Problema de exclusión mutua

- Disjkstra propuso en 1968 una solución usando **semáforos**.
- Un semáforo tiene dos operaciones indivisibles: **adquirir** y **liberar**.
- El código accedido de manera exclusiva suele denominarse **región crítica**.

DSIE (UPCT)

29

Problema de exclusión mutua

- Por ejemplo:

```
Adquirir ( semaforoAccesoDatosSensor )
Se accede a los datos del sensor
Liberar  ( semaforoAccesoDatosSensor )
```
- Es posible relajar la lectura a muchos lectores y un solo escritor.

DSIE (UPCT)

30

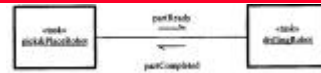
Problema de Sincronización de Tareas

- La **sincronización** se usa cuando dos tareas sincronizan sus operaciones sin intercambiar datos.
- La sincronización de eventos es **asíncrona**.
- La tarea origen ejecuta un **signal(event)**.
- La tarea destino ejecuta un **wait(event)** que interrumpe la tarea hasta recibir la señal, a no ser que ya se hubiera recibido.

DSIE (UPCT)

31

Ejemplo de Sincronización de Tareas



Pick & place robot:
While workAvailable **do**
 pick up part
 move part to work location
 release part
 move to safe location
signal (partReady)
wait (partCompleted)
 pick up part
 remove from work location
 place part
End while;

DSIE (UPCT)

drilling robot:
While workAvailable **do**
wait (partReady)
 move to work location
 drill four holes
 move to safe location
signal (partCompleted)
End while;

32

Problema Productor/Consumidor

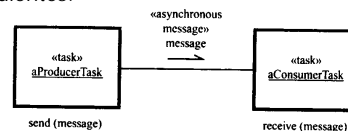
- En sistemas concurrentes, cada tarea tiene su propio hilo de control... **ejecución asíncrona**.
- El intercambio de datos necesita por tanto de sincronización.
- El productor debe producir antes de que el consumidor consuma!
- La comunicación puede ser síncrona o asíncrona.
- La síncrona puede ser con o sin respuesta.

DSIE (UPCT)

33

Comunicación Asíncrona

- El productor manda un mensaje y continúa su ejecución.
- El consumidor necesita una cola de mensajes pendientes dadas las diferentes velocidades.
- El consumidor se para si no hay mensajes pendientes.

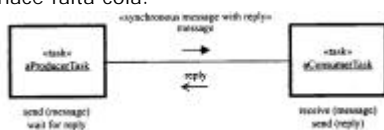


DSIE (UPCT)

34

Comunicación Síncrona con respuesta

- El productor manda un mensaje y espera la aceptación por parte del consumidor.
- Cuando el mensaje es aceptado se procesa, genera la respuesta y se envía al productor.
- Ambos continúan entonces en paralelo.
- No hace falta cola.

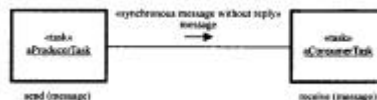


DSIE (UPCT)

35

Comunicación Síncrona sin respuesta

- El productor manda un mensaje y espera la aceptación por parte del consumidor.
- Cuando el mensaje es atendido ambos continúan en paralelo.
- No es necesaria cola alguna.



DSIE (UPCT)

36

Ejemplo de síncrona con respuesta



Vision system:

```

While workAvailable do
  wait (carArrived)
  take image of car body
  identify the model of car
  send carIDMess(data) to
    Robot system
  wait for reply
  ...
End while;
  
```

DSIE (UPCT)

37

Robot system:

```

While workAvailable do
  wait for mess. from
    Vision system
  receive carIDMess(data)
  execute ...
  send (doneReply) to
    Vision system
End while;
  
```

Monitores

- Un **monitor** es un objeto que encapsula datos y tiene operaciones para manipularlos de manera exclusiva.
- Un monitor combina ocultación de información con sincronización.
- Cada monitor tiene implícito un semáforo.
- Sólo una tarea activa por monitor.
- Las tareas deben esperar a que los monitores estén libres.

DSIE (UPCT)

38

Sincronización mediante Monitores

- Una tarea ejecutando en un monitor puede invocar **wait** hasta que se cumpla cierta condición.
- La tarea que se bloquea libera el bloqueo del monitor permitiendo a otras entrar.
- La tarea bloqueada será despertada por otra que haga **signal**.

DSIE (UPCT)

39

Tecnología para Sistemas Concurrentes y Distribuidos

DSIE (UPCT)

40

Tecnología para Sistemas Concurrentes y Distribuidos

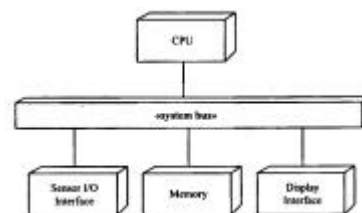
- ¿Cuáles son las infraestructuras habituales para las aplicaciones distribuidas y de tiempo real?

DSIE (UPCT)

41

Ambientes para el procesamiento Concurrente

- **Ambientes de Multiprogramación:** múltiples tareas comparten un procesador. La concurrencia es simulada.

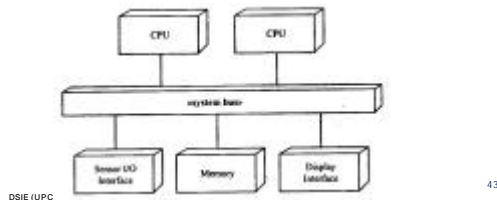


DSIE (UPCT)

42

Ambientes para el procesamiento Concurrente

- **Ambientes de Multiproceso Simétrico:** múltiples tareas comparten varios procesadores y una memoria. La concurrencia es real y la comunicación es vía la memoria compartida.

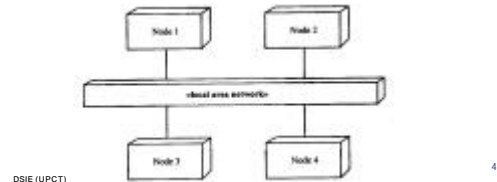


DSIE (UPC)

43

Ambientes para el procesamiento Distribuido

- Varios nodos están conectados a través de una red de comunicación. Cada nodo es un computador (de alguno de los anteriores). **No se comparte memoria!: la comunicación es mediante el envío de mensajes.**



DSIE (UPCT)

44

Soporte en ejecución

- **Núcleo del Sistema Operativo:**
 - ✦ Planificación de tareas por prioridades.
 - ✦ Comunicación entre tareas por paso de mensajes.
 - ✦ Semáforos para la exclusión mutua.
 - ✦ Sincronización de eventos usando señales.
 - ✦ Manejo de interrupciones y servicios I/O.
 - ✦ Manejo de memoria.
 - ✦ send message, receive message, wait y signal son primitivas del SO.
 - ✦ UNIX (Linux, Solaris, AIX), Windows 98/2000/NT, ...

DSIE (UPCT)

45

Soporte en ejecución

- **Núcleo del Sistema Operativo:**
 - ✦ **POSIX 1003 del IEEE** es un estándar del UNIX con extensiones para el procesamiento concurrente y de tiempo real:
 - ✦ Servicios para el manejo de tareas concurrentes.
 - ✦ Servicios de reloj.
 - ✦ Servicios de gestión de memoria.
 - ✦ Servicios I/O.

DSIE (UPCT)

46

Planificación de Tareas

- En sistemas de tiempo real la planificación de tareas por su prioridad suele ser lo más adecuado.



47

Cambio de contexto de Tareas

- **Ocurre:**
 - ✦ Cuando se bloquea a la espera de semáforo, mensaje o evento.
 - ✦ Cuando es "preempted" por otra de mayor prioridad.
- Se hace preciso guardar los registros Hw, el contador de programa, ...
- Posteriormente, cuando vuelva a ejecutarse se deberá restaurar el contexto guardado.

DSIE (UPCT)

48

La Entrada/Salida en los SOs

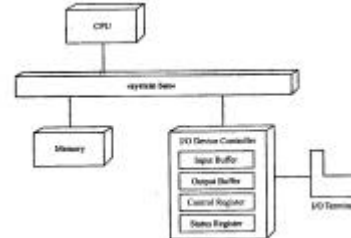
- Existen dos mecanismos generales para realizar I/O:
 - Interrupt driven I/O
 - Polled I/O
- Los dispositivos I/O se conectan con el sistema a través de **controladores de dispositivo I/O** (device controllers).
- Un controlador posee una serie de registros con los que se comunica la CPU.

DSIE (UPCT)

49

Controladores de Dispositivos

- El Sw que se ejecuta para acceder al controlador se llama **"device driver"** y es específico del dispositivo conectado.



DSIE (UPC)

50

Manejo de Interrupciones

- En el modo **"interrupt driven I/O"** se genera una interrupción cuando llega una entrada o se completa una salida.
- La llegada de una interrupción implica que la CPU haga un cambio de contexto para resolverla llamando al **manejador de interrupciones**.
- Este manejador determina la tarea que se debe activar para dicha interrupción.
- Así, el **"device driver"** es visto como una tarea más.

DSIE (UPCT)

51

Manejo de Interrupciones

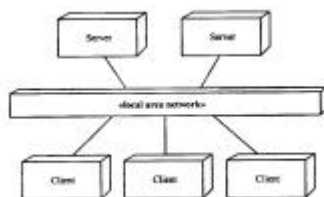
- En el modo **"polled I/O"** no hay interrupciones.
- El sistema muestrea periódicamente los dispositivos para determinar si ha llegado algún dato nuevo o si la salida depositada ha sido ya procesada.
- Los **"device drivers"** se convierten así en tareas periódicas.

DSIE (UPCT)

52

Tecnología para Sistemas Distribuidos

- Configuración **Cliente/Servidor**:
 - Clientes y Servidores conectados en red y geográficamente distribuidos.
 - Los clientes hacen peticiones a los servidores que procesan la respuesta y la devuelven al cliente.

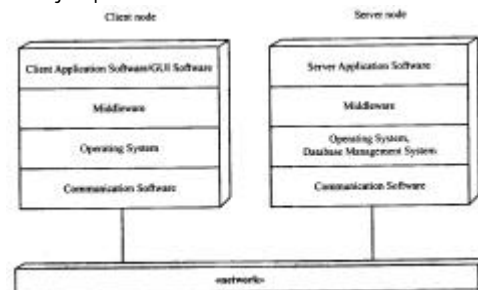


DSIE (UPCT)

53

Tecnología para Sistemas Distribuidos

- Ejemplo cliente/servidor:



54

Tecnología WWW

- La gran popularidad de la **World Wide Web** (1994, CERN) ha tenido una influencia decisiva en Internet.
- Una vista WWW para un usuario es a través del cliente que ejecuta local (Netscape, Explorer, ...).
- Las páginas WWW se mantienen en los servidores.

DSIE (UPCT)

55

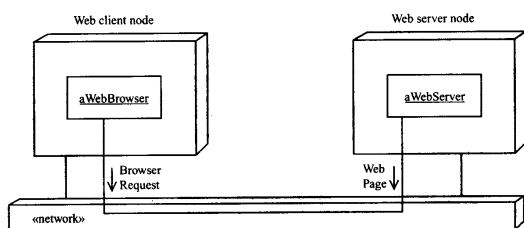
Tecnología WWW

- Las páginas WWW suelen editarse usando el lenguaje **HTML** o **XML**.
- Cada página se identifica mediante una **URL**.
- Un **plug-in** es un añadido en el **browser** del cliente de manera que extiende su capacidad (p.e. para Java, TCL/TK, ...).

DSIE (UPCT)

56

Tecnología WWW



DSIE (UPCT)

57

Tecnología WWW

- La evolución de los browsers ha llevado a que se desarrollen aplicaciones distribuidas usando la Web.
- El lenguaje de programación **Java**: www.java.sun.com
- Un **applet** Java es un programa Java descargado en un cliente desde un servidor.
- El browser interpreta el código del applet y lo ejecuta.
- Los programas Java ejecutados en el servidor se llaman **servlets**.
- La comunicación entre objetos Java distribuidos es a través de **RMI** (Remote Method Invocation).

DSIE (UPCT)

58

Servicios de los SOs distribuidos

- Un **Sistema Distribuido (SD)** está formado por computadores conectados por un medio de comunicación como p.e. una red de área local.
- **Servicio de Nombres**:
 - Transparencia de localización.
 - **DNS** (Domain Name System).
 - Cada nodo tiene una IP (32 bits): **128.174.40.15**

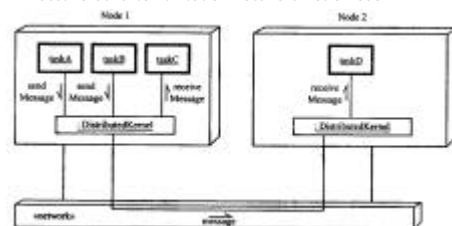
DSIE (UPCT)

59

Servicios de los SOs distribuidos

- Servicios de Comunicación de **Mensajes Distribuidos**:

- El kernel local busca en una tabla dónde reside la tarea destino de la comunicación: local o en otro nodo.



DSIE (UPCT)

60

Servicios de los SOs distribuidos

- Servicios de **Sockets**:
 - ✦ Los proveen los SOs.
 - ✦ Proveen un cjo. de operaciones que una aplicación puede invocar cuando se comunica con otra aplicación en una red.
 - ✦ Usa TCP/IP.

DSIE (UPCT)

61

Servicios de los SOs distribuidos

- Comunicación a través de **puertos**:
 - ✦ A veces el destinatario no se identifica por el nombre sino que tiene asociado un puerto.
 - ✦ En la configuración del sistema se asigna el puerto a un destino específico.
 - ✦ Mayor flexibilidad.
- **Recuperación** de errores:
 - ✦ Un error de paridad implica que la red retransmita el mensaje.
 - ✦ Si el destino no recibe el mensaje, la red debe notificar al cliente que no se recibió.
 - ✦ Tanembaun[1992] y Bacon[1997].

DSIE (UPCT)

62

Servicios de los SOs distribuidos

- **Middleware**:
 - ✦ Muchas veces el entorno es heterogéneo en Hw y Sw (Windows, Unix, Pc. Workstations, ...).
 - ✦ Middleware es una capa sobre el SO que presenta una plataforma uniforme sobre la que las aplicaciones distribuidas pueden ejecutarse.
 - ✦ RPC (Remote Procedure Call) fue una versión primitiva del middleware.
 - ✦ CORBA, COM, Java-RMI.

DSIE (UPCT)

63

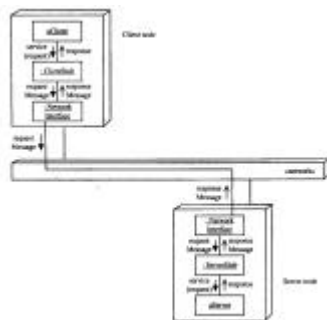
Servicios de los SOs distribuidos

- **Remote Procedure Call (RPC)**:
 - ✦ Un cliente en un nodo hace una llamada remota a un servidor en otro nodo.
 - ✦ El mensaje se empaqueta y se envía al servidor.
 - ✦ El servidor desempaqueta el mensaje y llama al procedimiento apropiado en él.
 - ✦ El resultado se envía al cliente.

DSIE (UPCT)

64

Servicios de los SOs distribuidos



DSIE (UPCT)

65

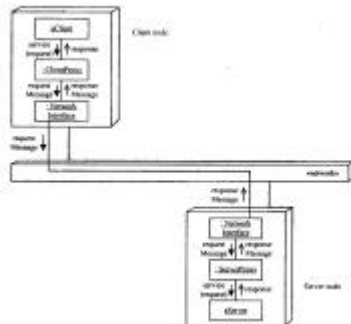
Servicios de los SOs distribuidos

- **Java-RMI**:
 - ✦ El JDK soporta un **middleware** llamado **RMI** que permite a objetos distribuidos comunicarse entre sí.
 - ✦ No se manda un mensaje a un procedimiento específico como hace RPC.
 - ✦ El cliente envía el mensaje a un objeto específico e invoca un método del mismo.
 - ✦ El **client-proxy** provee al cliente el mismo interfaz como si fuera el servidor, escondiendo todos los detalles relativos a la comunicación.
 - ✦ En el servidor, un **server-proxy** invoca el método del servidor. Si el objeto servidor no existe, entonces lo crea.
 - ✦ La elección de qué servidor provee el servicio se resuelve en tiempo de ejecución.

DSIE (UPCT)

66

Servicios de los SOs distribuidos: Java RMI



DSIE (UPC1)

67

Servicios de los SOs distribuidos

CORBA:

- ✦ Estándar para sistemas abiertos desarrollado por el **OMG** (Object Management Group).
- ✦ Permite la comunicación entre objetos sobre plataformas heterogéneas.
- ✦ El middleware se llama **ORB** (Object Request Broker).
- ✦ Con el ORB, el cliente solicita un servicio a un objeto sin saber dónde está ubicado ni plataforma, ni protocolo de comunicación, ni lenguaje, etc.
- ✦ El interfaz de los objetos se especifica utilizando el lenguaje **IDL** (Interface Definition Language) de CORBA.

DSIE (UPCT)

68

El Ciclo de Desarrollo de Software

DSIE (UPCT)

69

Modelos de Proceso Software

- ✦ En 1969 el término Ingeniería del Sw se acuñó para referirse al manejo y la definición de métodos y técnicas, procedimientos y herramientas para el desarrollo de Sw a gran escala.
- ✦ Estos proyectos grandes se desarrollaban siguiendo una serie de fases en un orden prefijado.

DSIE (UPCT)

70

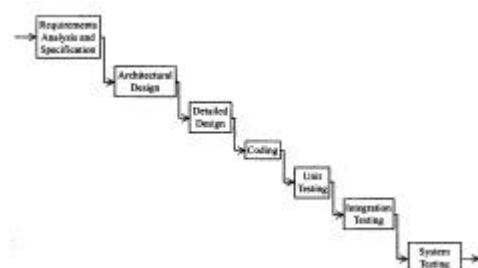
Modelos de Proceso Software

- ✦ El más clásico fue el **modelo en cascada** (waterfall):
 - ✦ Es fácil determinar en qué situación se está del proyecto.
 - ✦ Se genera documentación clara en cada fase.
- ✦ Pero...
 - ✦ Los requisitos no siempre están claros desde el principio.
 - ✦ Los errores cometidos se detectan muy tarde.

DSIE (UPCT)

71

Modelo en Cascada



DSIE (UPCT)

72

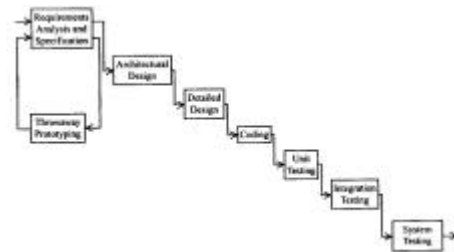
Modelos de Proceso Software

- La **prototipación** resuelve algunos de los problemas anteriores.
 - » Los requisitos se clarifican mejor.
 - » El interfaz queda bien definido.
- Los prototipos evolutivos llegan a ser el producto final.
- Los prototipos desechables se abandonan, sólo sirven para clarificar los requisitos.
- Pero,
 - » Requieren una inversión adicional (usuarios más involucrados, herramientas específicas, ...)
 - » El usuario cree muy pronto tener algo que no existe.

DSIE (UPCT)

73

Prototipación



DSIE (UPCT)

74

Prueba del Software

- Gran parte de las técnicas de prueba del Sw concurrente y de tiempo real son las mismas que para el resto de sistemas.
- Las mayores diferencias radican en el hecho de que el Sw está constituido por tareas concurrentes y que el sistema interactúa con muchos dispositivos externos.

DSIE (UPCT)

75

Prueba del Software

- Un problema con la prueba de Sw concurrente es que **la ejecución es no determinista**.
- Pruebas unitarias.
- Pruebas de Integración.
- Pruebas del sistema para comprobar que se cumplen los requisitos:
 - » Pruebas de la Funcionalidad.
 - » Pruebas de carga del sistema.
 - » Pruebas de tiempo de respuesta.
- Pruebas de aceptación.

DSIE (UPCT)

76

Evolución de los Métodos de Diseño Software

- Años 60: no sistemática. Flowcharts...
- Ppos. 70: Programación Estructurada.
- 1972: Principio de Ocultación de Información (David Parnas).
- Mediados 70: Análisis y Diseño Estructurado.
- **Años 80: Sistemas en Concurrentes y de Tiempo Real.**

DSIE (UPCT)

77

Detalle de la Metodología COMET

DSIE (UPCT)

78

COMET

- **COMET** (Concurrent Object Modeling and architectural design mEThod).
- COMET es un proceso de desarrollo de Sw orientado a objetos y compatible con el modelo de proceso de **UML** (Unified Software Development Process).

DSIE (UPCT)

79

COMET

- Es cíclico, iterativo y gira alrededor de los **casos de uso**.
- Los requisitos funcionales del sistema se definen en términos de **actores** y **casos de uso**.
- Cada caso de uso define una secuencia de interacciones entre uno o más actores y el sistema.

DSIE (UPCT)

80

COMET

- Un caso de uso se puede ver a distintos niveles de detalle.
- En el *modelo de requisitos* los requisitos funcionales del sistema se definen en términos de actores y casos de uso.
- En el *modelo de análisis* el caso de uso se refina para describir objetos que participan en el caso de uso y sus interacciones.

DSIE (UPCT)

81

COMET

- En el *modelo de diseño* se desarrolla la arquitectura del sistema fijando aspectos de distribución, concurrencia.

DSIE (UPCT)

82

Modelo de Requisitos

- Se desarrolla una descripción narrativa de cada caso de uso.
- Cada caso de uso es visto como una caja negra.
- Si los requisitos no se entienden se puede utilizar un prototipo desechable.
- Objetivo:

definición de los requisitos
funcionales del sistema

DSIE (UPCT)

83

Modelo de Análisis

- Se desarrollan modelos estáticos y dinámicos del sistema.
- Los modelos estáticos definen las relaciones estructurales entre las clases del dominio del problema.

DSIE (UPCT)

84

Modelo de Análisis

- Los modelos dinámicos se desarrollan refinando los casos de uso.
- Objetivo:

conocimiento del
dominio del problema

DSIE (UPCT)

85

Actividades del Análisis

- Modelado Estático:**
 - Es una vista estructural del sistema.
 - Clases + atributos + relaciones entre clases.
 - Se modelan clases del dominio del problema
clases reales!
- Estructuración de objetos:**
 - Objetos de cada caso de uso.
 - Objetos Entidad, interfaz, de control o de la lógica de la aplicación.

DSIE (UPCT)

86

Actividades del Análisis

- Modelado Máquina de Estados:**
 - Cada objeto dependiente del estado tendrá un diagrama de estados asociado.
- Modelado Dinámico:**
 - Los casos de uso se refinan.
 - La interacción entre los objetos de los casos de uso se refleja en diagramas de interacción.

DSIE (UPCT)

87

Modelo de Diseño

- Se desarrolla la **arquitectura** del sistema.
- Del **dominio del problema** se pasa al **dominio de la solución**.
- El modelo de diseño considera la concurrencia del sistema.

DSIE (UPCT)

88

Actividades del Diseño

- Sintetizar los **diagramas de colaboración** del análisis en **uno único** para el diseño.
- Dar una arquitectura del sistema en forma de **subsistemas** y sus **interfaces**.
- Encontrar objetos activos (**tareas**).
- Estudiar la **sincronía** de los **mensajes**.
- Estudiar la **interacción** entre tareas.

DSIE (UPCT)

89

Actividades del Diseño

- Analizar la productividad** del modelo de diseño a nivel de tarea y de subsistema.
 - Aplicar las técnicas de planificación de tareas para tiempo-real.
- Diseñar detalladamente las clases de cada subsistema.

DSIE (UPCT)

90

Construcción de Sw incremental

- Después del diseño arquitectónico se construye el Sw de manera incremental.
- En cada incremento se construye un subsistema.

DSIE (UPCT)

91

Integración de Sw incremental

- Se hacen pruebas de integración incrementales basándose en los casos de uso.
- Pruebas del sistema (caja negra).

DSIE (UPCT)

92

Diagramas por fase

- **Captura de requisitos:**
 - ✦ Diagramas de Casos de Uso.
 - ✦ Documentación de cada diagrama.
- **Análisis:**
 - ✦ Diagramas de Clases del Análisis.
 - ✦ Diagrama de Contexto.
Imprescindible cuando hay dispositivos y sistemas externos!
 - ✦ Diagramas de Interacción (Colaboración o Secuencia).
 - ✦ Diagramas de Transición de Estados.

DSIE (UPCT)

93

Diagramas por fase

- **Diseño:**
 - ✦ D. de Transición de Estados consolidado.
 - ✦ D. de Colaboración consolidado.
Particionado en subsistemas si el sistema es grande!
 - ✦ D. de Clases consolidado.

DSIE (UPCT)

94

Contenido Parte II

- Fase de Análisis en COMET
 - ✦ Modelado de Casos de Uso
 - ✦ Modelado Estático
 - ✦ Estructuración en Clases y Objetos
 - ✦ Diagramas de Transición de Estados
 - ✦ Modelado Dinámico

DSIE (UPCT)

95