Diseño Arquitectónico

- Establecer la arquitectura global del sistema de software

Objetivos

- Introducir el diseño arquitectónico y discutir su importancia
- Explicar por qué se requieren múltiples modelos par documentar una arquitectura de software
- Describir diferentes tipos de modelos de arquitectura que puede ser usados
- Discutir cómo modelos de referencia específicos del dominio pueden ser usado como una base para líneas de productos y para comparar arquitecturas de software

Tópicos cubiertos

- Estructuración del sistema
- Modelos de control
- Descomposición modular
- Arquitecturas específicas de dominio

Arquitectura de software

- El *diseño arquitectónico* corresponde al proceso de diseño que identifica los subsistemas que conforman un sistema y la infraestructura de control y comunicación
- La salida de este proceso de diseño es una descripción de la *arquitectura de software*

Diseño arquitectónico

- Etapa temprana del proceso de diseño del sistema
- Representa el puente entre el proceso de especificación y diseño
- A menudo se ejecuta en paralelo con algunas actividades de especificación
- Involucra la identificación de los componentes principales del sistema y su comunicación

Ventajes de arquitectura explícita

- Comunicación entre los Stakeholders
  - La arquitectura puede ser usada como un foco de discusión por los stakeholders del sistema
- Análisis de sistemas
  - Ayuda a establecer si el sistema puede cumplir los requerimientos no funcionales.
- Reutilización a gran escala
  - La arquitectura puede ser reutilizada a través de un rango de sistemas

Proceso de diseño arquitectónico

- Estructuración del sistema
  - El sistema se descompone en varios subsistemas principales y la comunicación entre estos subsistemas es identificada
- Modelado del control
  - Se establece un modelo de las relaciones de control entre las diferentes partes del sistema
- Descomposición modular
  - Los subsistemas identificados se descomponen en módulos

Subsistemas y módulos

- Un subsistema es un sistema por derecho propio cuya operación es independiente de los servicios provistos por otros subsistemas
- Un módulo es un componente del sistema que provee servicios a otros componente pero no se consideraría normalmente como un sistema separado

Modelos arquitectónicos

- Diferentes modelos arquitectónicos pueden ser producidos durante el proceso de diseño
- Cada modelo presenta diferente perspectivas de la arquitectura

Modelos arquitectónicos

- Modelo estático estructurales que muestra los componentes principales del sistema
- Modelo dinámico del proceso que muestra la estructura de proceso del sistema
- Modelo de interfaz que define las interfaces de los subsistemas
- Modelo de relaciones tales como un modelo de flujo de datos

Estilos arquitectónicos

- El modelo arquitectónico de un sistema podría estar en conformidad con un modelo más genérico o estilo
- Estar al tanto de estos estilos puede simplificar el problema de definir arquitecturas de sistemas
- Sin embargo, la mayoría de sistemas grandes son heterogéneos y no siguen un único estilo arquitectónico

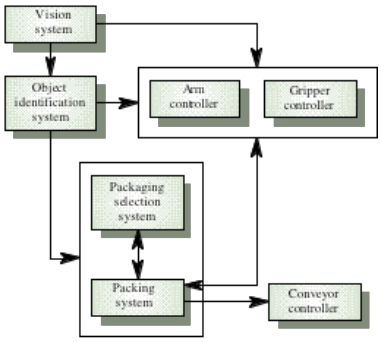
Atributos de la arquitectura

- Desempeño
  - Localizar operaciones para minimizar la comunicación entre subsistemas
- Seguridad
  - Usar una arquitectura de capas con recursos críticos en capas internas
- Protección
  - Aislar componentes componentes críticos de seguridad
- Disponibilidad
  - Incluir componentes críticos en la arquitectura
- Mantenibilidad
  - Usar componentes autocontenidos de grano fino

Estructuración del sistema

- Concerniente con la descomposición del sistema en subsistemas que interactúan
- El diseño arquitectónico se expresa normalmente como un diagrama de bloques que representa un visión general de la estructura del sistema
- Se pueden desarrollar modelos más específicos que muestran cómo los subsistema comparten datos, cómo se distribuyen y cómo se comunican entre si.

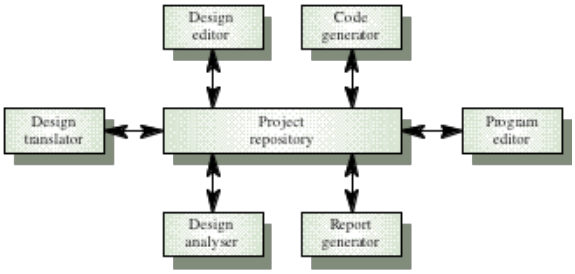
Sistema de control de robot empacador



Modelo de depósito

- Los subsistemas deben intercambiar datos. Esto puede ser hecho de dos formas:
  - Los datos compartidos se mantiene en una base de datos central o depósito y puede ser accedida por todos los subsistemas
  - Cada subsistema mantiene su propia base de datos y pasa datos explícitamente a otros subsistemas
- Cuando grandes cantidades de datos deben ser compartidos, el modelo de depósito es el más comúnmente usado

Arquitectura de una herramienta CASE



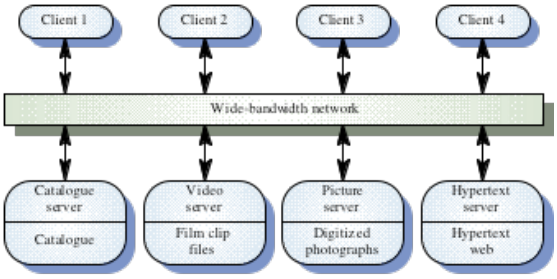
Características del modelo de depósito

- Ventajas
  - Forma eficiente de compartir grandes cantidades de datos
  - Los subsistemas no se deben preocupar sobre cómo los datos son producidos o usados
  - Administración centralizada. Ej. Backup, seguridad
  - El modelo de compartición es visible a lo largo del esquema de depósito
- Desventajas
  - Los subsistemas deben acordar un modelo de datos del depósito. Lo cual es inevitablemente un compromiso
  - La evolución de datos es difícil y cara
  - No hay campo para políticas de administración específicas
  - Es difícil distribuir el depósitos eficientemente

Arquitectura de cliente-servidor

- Modelo de sistema distribuido el cual muestra cómo los datos y el procesamiento se distribuyen a través de un rango de componentes
- Conjunto de servidores stand-alone que proveen servicios específicos tales como impresión, administración de datos, etc.
- Conjunto de clientes los cuales acceden a estos servicios
- Una red la cual permite la comunicación entre clientes y servidores

Biblioteca de videos y pinturas



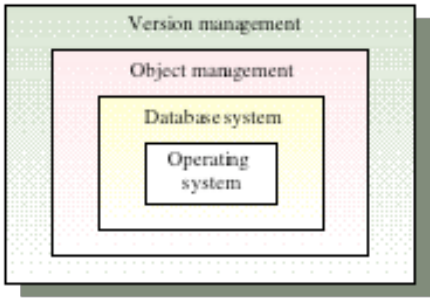
Características del modelo cliente-servidor

- Ventajas
  - La distribución de datos es directa
  - Hace uso efectivo de sistemas interconectados. Podría requerir hardware más barato
  - Es fácil adicionar nuevos servidores o actualizar servidores existentes
- Desventajas
  - No hay un modelo de datos compartido, de manera que los subsistemas usan una organización de datos diferente. El intercambio de datos puede ser ineficiente
  - Administración redundante en cada servidor
  - No hay un registro central de nombres y servicios

Modelo de máquina abstracta

- Usado para modelar las interfaces en entre subsistemas
- Organiza el sistema en un conjunto de capas (o máquinas abstractas) cada una de la cuales provee un conjunto de servicios
- Soporta el desarrollo incremental de subsistemas en diferentes capas. Cuando la interfaz de una capa cambia, solo las capas adyacentes son afectadas
- Sin embargo, es difícil, en general, estructurar sistemas de esta forma

Sistema de manejo de versiones



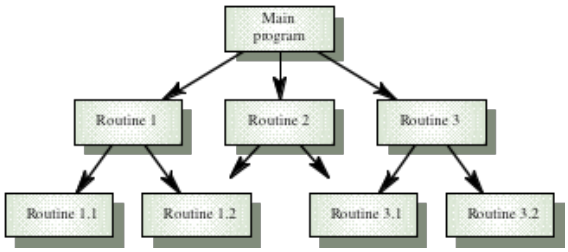
Modelos de control

- Concernientes con el flujo de control entre subsistemas. Distintos al modelo de descomposición del sistema.
- Control centralizado
  - Un subsistema tiene responsabilidad general de controlar y comenzar otros subsistemas
- Control basado en eventos
  - Cada subsistema puede responder a eventos generados externamente desde otros subsistemas o el entorno del sistema

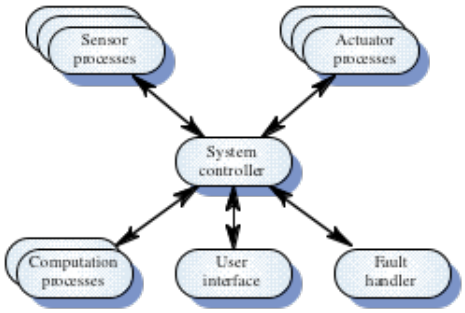
Control centralizado

- Un subsistema tiene responsabilidad general de controlar y comenzar otros subsistemas
- Modelo de llamada-retorno
  - Modelo de subrutina top-down donde el control comienza en el tope de una jerarquía de subrutinas y se mueve hacia abajo. Aplicable a sistemas secuenciales
- Modelo de administrador
  - Aplicable a sistemas concurrentes. Un componente del sistema controla la parada, arranque y coordinación de otros procesos del sistema. Puede ser implementado en sistemas secuenciales como una instrucción switch-case

Modelo de llamada retorno



Sistema de control de tiempo real



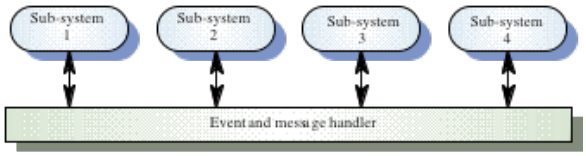
Sistemas dirigidos por eventos

- Dirigidos por eventos generados externamente dodne el tiempo de el evento está fuera del control de los subsistemas que procesan el evento
- Tipos:
  - Modelos broadcast. Una evento se transmite a todos los subsistema. El evento puede ser procesado por cualquier subsistema que esté en capacidad de hacerlo.
  - Modelos dirigidos por interrupciones. Usados en sistemas de tiempo real donde las interrupciones son detectadas por un manejador de interrupciones y pasadas a otros componente para ser procesadas.
- Otros modelos dirigidos por eventos incluyen hojas electrónicas y sistemas de producción

Modelo de broadcast

- Efectivo para integrar subsistema en diferentes computadores en una red
- Los subsistemas registran interés en eventos específicos. Cuando esto ocurre, el control se transfiere a los subsistemas que pueden manejar el evento.
- La política de control no se embebe en el manejador de eventos y mensajes. Los subsistemas deciden sobre eventos de interés para ellos
- Sin embargo, los subsistemas no conocen si un evento va ser procesado o no

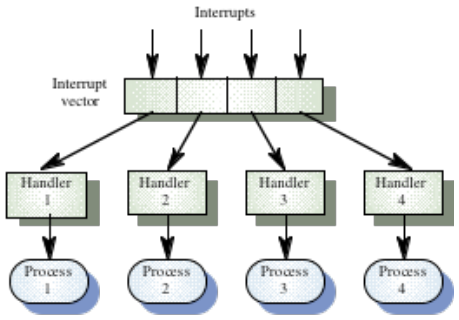
Selective broadcasting



Sistemas dirigidos por interrupciones

- Usados en sistemas de tiempo real donde una respuesta rápida a un evento es esencial
- Hay diferentes tipos de interrupciones con un manejador definido para cada tipo
- Cada tipo está asociado con una posición de memoria y un interruptor de hardware causa la transferencia a este manejador
- Permite una rápida respuesta, pero son complejos de programar y difíciles de validar

Interrupt-driven control



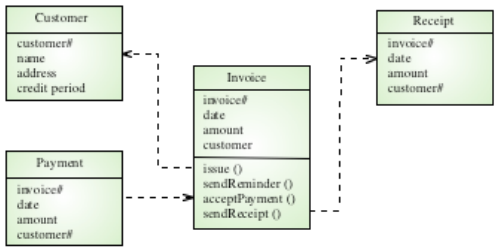
Descomposición modular

- Un nivel estructural adicional donde los subsistemas son descompuestos en módulos
- Dos modelos de descomposición modular cubiertos:
  - Un modelo de objetos donde el sistema es descompuesto en objetos que interactúan
  - Un modelo de flujo de datos donde el sistema es descompuesto en módulos funcionales los cuales transforman entradas en salidas. Conocido también como modelo de tubería (pipeline)
- Si es posible, las decisiones acerca de la concurrencia deben ser dejadas para la etapa de implementación

Modelos de Objetos

- Estructura el sistema en un conjunto de objetos débilmente acoplados con interfaces bien definidas
- La descomposición orientada a objetos es concerniente con la identificación de clases de objetos, sus atributos y sus operaciones
- En la implementación, los objetos se crean a partir de estas clases y un modelo de control determinado es usado para coordinar la operaciones de los objetos

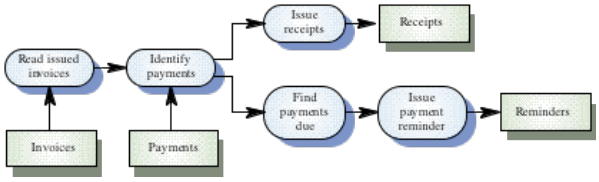
Invoice processing system



Modelos de flujos de datos

- Transformaciones funcionales procesan sus entradas para producir salidas
- Este modelo se puede ver como un modelo de filtros y tuberías
- Variaciones de este modelo son bastante comunes. Cuando las transformaciones son secuenciales, se habla de de modelo de proceso de lotes secuencial, el cual es usado extensivamente en sistemas de procesamiento de datos
- No se adecua muy bien para sistemas interactivos

Invoice processing system



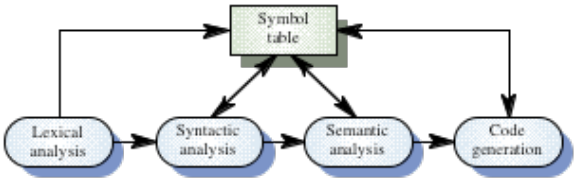
Arquitecturas específicas del dominio

- Modelos arquitectónicos específicos a un dominio de aplicación
- Dos tipos de modelos de dominio específico:
  - Modelos genéricos los cuales son abstracciones de un número de sistemas reales y que encapsulan las características principales de estos sistemas
  - Modelos de referencia los cuales son más abstractos. Modelos idealizados. Proveen un medio de información acerca de cierta clase de sistemas y permiten comparar diversas arquitecturas.
- Los modelos genéricos son usualmente modelos bottom-up; los modelos de referencia son modelos top-down

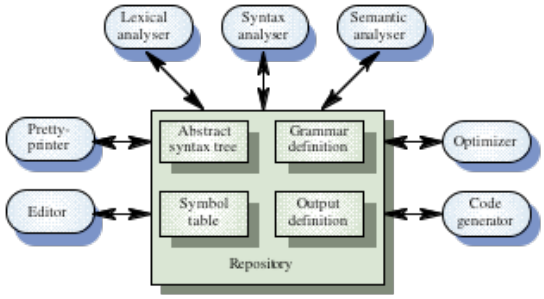
Modelos genéricos

- El modelo de un compilador es un ejemplo bien conocido, sin embargo existen otros modelos en dominios de aplicación más especializado:
  - Analizador léxico
  - Tabla de símbolos
  - Analizador sintáctico
  - Árbol de sintaxis
  - Analizador semántico
  - Generador de código
- El modelo genérico de compilador podría ser organizado de acuerdo a diferentes modelos arquitectónicos

Modelo de Compilador



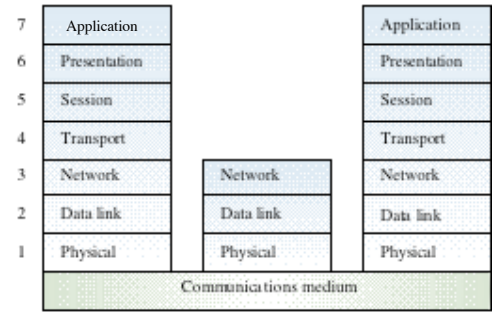
Sistema de procesamiento de lenguaje



Arquitecturas de referencia

- Los modelos de referencia se derivan a partir de un estudio del dominio de la aplicación antes que a partir de sistemas existentes
- Pueden ser usado como una base para la implementación del sistema o para comparar diferentes sistemas. Actúan como un estándar para evaluar sistemas.
- El modelo OSI es un modelo en capas para sistemas de comunicación

Modelo de referencia OSI



Puntos claves

- La responsabilidad del arquitecto de software es derivar un modelo estructural del sistema, un modelo de control y modelo de descomposición en subsistemas
- Los sistemas grandes raramente se acomodan a un modelo arquitectónico simple
- Los modelos de descomposición del sistema incluyen: modelos de depósito, modelos cliente-servidor y modelos de máquina abstracta
- Los modelos de control incluyen control centralizado y modelos dirigidos por eventos

Puntos claves

- Los modelos de descomposición incluyen modelos de flujo de datos y modelos de objetos
- Los modelos arquitectónicos son abstracciones sobre un dominio de aplicación. Pueden ser construidos abstrayendo modelos existentes o pueden ser modelos de referencia idealizados