

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
CENTRO UNIVERSITARIO UAEM ATLACOMULCO
INGENIERÍA EN COMPUTACIÓN

DISEÑO DE SISTEMAS

TAREA # 7

QUE PRESENTA:

DAVID MARTÍNEZ PEÑA

ICO 9

Atacomulco, México a 17 de Noviembre de 2008.

EL MODELO DE IMPLANTACION DEL USUARIO

El modelo esencial describe:

La política o lógica esencial de las funciones que se requiere realizar

El contenido esencial de los datos que almacena el sistema y que se mueven a través de el.

El comportamiento esencial dependiente del tiempo que el sistema debe exhibir para manejar señales e interrupciones del ambiente exterior.

El formato de las entradas y salidas del sistema es de enorme interés para el usuario. La importancia de estas cuestiones de implantación es:

¿Qué partes del modelo esencial se asignarán a la PC y que partes a la computadora principal?, ¿Qué partes de los datos se asignará a la PC y cuál a la principal?, ¿Qué parte de los datos se asignara a la PC y cual a la principal?, ¿Qué formato tendrán las entradas que el usuario proporciona a la PC?, ¿Qué actividades adicionales de apoyo hay que proporcionar para asegurar que el usuario no dañe inadvertidamente los datos almacenados en la PC o en la computadora principal?

El analista y el usuario podrían hacer un prototipo de porciones del sistema utilizando un lenguaje de cuarta generación o un paquete de generación de aplicaciones.

En muchas aplicaciones de negocios, otra opción para el usuario es la selección y compra de un paquete de software, es decir, un producto ya existente que puede comprarse o usarse bajo licencia.

DETERMINACION DE LA FRONTERA DE AUTOMATIZACION

Hay 3 casos extremos:

1. Al usuario pudiera no importarle dónde esté la frontera de automatización.
2. El usuario podría escoger un sistema totalmente automatizado.
3. El usuario podría optar por un sistema completamente manual.

El analista debe reconocer las actividades manuales que son parte del nuevo sistema, por ello pueden tener que escribir procedimientos para los usuarios de modo que sepan como llevar a cabo las funciones requeridas, además de proporcionar alguna guía sobre la organización de los almacenes que no se van a automatizar.

DETERMINACION DE LA INTERFAZ HUMANA

La actividad que consume más tiempo y que más interesa al usuario es la especificación de la interfaz humana, esto involucra 4 puntos:

- La elección de los dispositivos de entrada y salida.
- El formato de todas las entradas que fluyen desde los terminadores hasta el sistema.
- El formato de las salidas que fluyen desde el sistema hacia los terminadores.
- La secuencia y los tiempos de entradas y salidas en un sistema en línea.

DISPOSITIVOS DE ENTRADA Y SALIDA

Los dispositivos que se usan típicamente para proporcionar entradas al sistema incluyen los siguientes:

- *Tarjetas perforadas*: una ventaja es que puede ser utilizada como documento fuente por el usuario externo. Las principales desventajas son su tamaño estorbo, su limitada capacidad de almacenamiento de dato, el hecho que solo pueden usarse y la susceptibilidad a errores del operador.
- *Cinta magnética*: la principal ventaja es que se puede almacenar un volumen mucho mayor de datos en una cinta que en una tarjeta, la desventaja es que los datos no se pueden manipular fácilmente una vez que se graban en la cinta.
- *Discos flexibles*: puede almacenar entre 360,000 y 1.2 millones de caracteres, esto no es tanto como lo que almacena una cinta magnética.
- *Terminadores y computadoras personales*: las terminales inteligentes y las PC vuelven posible que el usuario haga cambios y corrija errores triviales de manera instantánea, en lugar del retraso de mandar las entradas mediante líneas de telecomunicaciones a una computadora principal.
- *Lectores ópticos y lectores de código de barras*: algunos lectores ópticos pueden leer documentos ordinarios mecanografiados y algunos pueden leer documentos manuscritos, la desventaja es su costo y su tendencia a los errores.
- *Teléfono*: el teléfono por tonos puede ser un medio apropiado de ingreso de entradas.
- *Voz*: algunos sistemas pueden usar la voz humana como medio de entrada, las desventajas son: el costo del dispositivo, su limitado vocabulario, su tiempo de respuesta es lento, los verdaderos problemas si la voz del usuario cambia de manera significativa debido a un resfriado o alguna otra causa.

MEDIOS PARA SALIDAS

- *Salidas impresas*: la principal ventaja de las salidas impresas es que constituye un documento permanente y puede usarse en una variedad de aplicaciones fuera del sistema, las desventajas de los reportes impresos son su tamaño estorbo, la probabilidad de que se impriman más copias de lo que realmente se ocupa y la velocidad relativamente baja a la que se produce la información.
- *Tarjetas perforadas*
- *Termina*: la ventaja es que puede mostrar una gran cantidad de información a gran velocidad, su desventaja es que no representa una salida material, la salida es transitoria y se pierde cuando se muestra el siguiente desplegado.
- *Salida de voz*: la principal ventaja es que se puede usar para comunicar mensajes relativamente breves en un medio, donde el usuario no tenga oportunidad de leer salidas impresas.

- *Graficador*: sirve para producir diagramas y dibujos grandes y complejos, la desventaja es su tamaño y la cantidad de tiempo que se requiere para producir sus salidas.
- *Cinta magnética o disco*
- *COM (Microforma para salidas de computadora)*: se reservan para archivos, los rollos de microfilm o las tarjetas de microfichas son ejemplos de esto.

FORMATOS DE ENTRADA Y SALIDA

1. El sistema debe pedir entradas y producir salidas en forma consistente.
2. Pida información con una secuencia lógica.
3. Haga obvio al usuario el tipo de error que ha cometido y dónde: cada error debe identificarse con un mensaje descriptivo o bien enfatizando o mostrando el dato erróneo de acuerdo con algún código de colores.
4. Distinga entre edición de campos y ediciones de pantalla: el sistema será capaz de determinar si el dato proporcionado por el usuario es o no correcto sin hacer referencia a otros datos.
5. Haga la edición y la revisión de errores dependiente del usuario.
6. Permita que el usuario pueda: cancelar parte de la transacción y cancelarla toda.
7. Proporcione un mecanismo de ayuda conveniente.
8. Distinga entre sistemas guiados por menús y sistemas dirigidos por órdenes; si es apropiado, déle a escoger al usuario: un sistema dirigido por menús representa una lista de opciones alternativas. Un sistema dirigido por ordenes espera que el usuario proporcione una directiva detallada indicando lo que el sistema quiere que haga por el.
9. Si el sistema está realizando un proceso largo, despliegue un mensaje al usuario para que no crea que se detuvo.
10. Proporcione alternativas por omisión para las entradas estándar.
11. Aproveche el color y sonido pero no abuse.

DISEÑO DE LAS FORMAS

Es el diseño de los formatos de entrada y salida de un sistema, debe contener la siguiente información básica:

- **Título**: se imprime con letras grandes y resaltadas en la parte superior de la forma para que el usuario sepa que es una “forma de pedido” o bien una “forma de reporte de fallas”
- **Instrucciones**: se colocan al principio de la forma, donde se ingresan los datos.
- **Cuerpo**: es la parte principal de la forma, donde se ingresan los datos.

LOS TIPOS DE FORMAS DE ESPECIALIDAD SON:

- Formas empastadas en libros.
- Formas múltiples desprendibles con un original y varias copias que se pueden separar.
- Formas continuas que se llenan manualmente o por computadora.
- Para correo: formas preimpresas insertas en un sobre, unidas como en una forma continua.

CODIGOS DE ENTRADA Y SALIDA

Como parte de la labor de especificar formatos de entrada y salida, el analista muchas veces debe especificar códigos, es decir, abreviaciones de la información que sería difícil y tardado describir con detalle. Un método de codificación debe ser:

1. **Expandible**: Debe proporcionar espacio para entradas adicionales que pudieran requerirse.
2. **Preciso**: Debe identificar al artículo específico.
3. **Conciso**: Debe ser breve pero describir adecuadamente el artículo.
4. **Conveniente**: Debe ser fácil de codificar y decodificar.
5. **Con significado**: Debe ser útil para quienes lo manejan, de ser posible debe indicar algunas de las características del artículo.
6. **Operable**: Debe ser compatible con los métodos presentes y anticipados de proceso de datos, manual o a máquina.

IDENTIFICACION DE LAS ACTIVIDADES DE APOYO MANUAL ADICIONAL

- Ingreso de datos al sistema
- Realización de los cálculos
- Almacenamiento a largo plazo de los datos
- Salidas de datos del sistema

Podría decidirse añadir cualquiera de lo siguiente al modelo esencial para vérselas con la tecnología defectuosa:

- Entradas o salidas redundantes: Habría duplicados de las entradas desde dos fuentes distintas.
- Tecnología de procesador redundante: Los datos que el sistema mantiene pueden almacenarse por duplicado, en dos discos o cintas diferentes.
- Redundancia interna
- Controles por lote (batch): el sistema puede mantener su propia cuenta del número de transacciones que ha dado como salidas para luego compararlas con una cuenta manual proporcionada por el usuario.
- Verificaciones secuenciales: el usuario asigna un número de secuencia o de transacción a cada entrada y el sistema verifica al procesarlas que estén en secuencia y que no se pierda ninguna transacción.

ESPECIFICACION DE RESTRICCIONES OPERACIONALES

El equipo de implantación tendrá que decidir la combinación de hardware, el sistema operativo, equipo de telecomunicaciones, lenguaje de programación y estrategia de diseño para implantar mejor los requerimientos. Las cuestiones típicas son:

1. Volumen de datos: el usuario necesita especificar los volúmenes de transacciones de entrada y el tamaño requerido de los almacenes de datos.
2. Tiempo de respuesta a las diversas entradas.
3. Restricciones políticas sobre modalidades de implantación: el usuario pudiera especificar la marca de hardware que se usará, el lenguaje de programación, los proveedores de telecomunicaciones que se usarán.
4. Restricciones ambientales: el usuario que trabaja en el equipote implantación pudiera imponer restricciones de temperatura, humedad, interferencia eléctrica, consumo de energía, limitaciones de tamaño, peso o emisiones eléctricas, vibración, contaminación, ruido, radiación y otras restricciones ambientales.
5. Restricciones de seguridad y confiabilidad: el usuario pudiera especificar un tiempo promedio entre fallas y tiempo promedio necesario para la reparación para el sistema.
6. Restricciones de seguridad: el usuario puede incluir mecanismo para evitar el acceso no autorizado a datos confidenciales. El usuario podría solicitar mecanismos para evitar que usuarios no autorizados realicen ciertas funciones en el sistema. Podrían imponer diversas medidas de seguridad a los datos que entran o salen del sistema.

DISEÑO

Las etapas del diseño: Los modelos más importantes para el diseñador son el modelo de implantación de sistemas y el modelo de implantación de programas. El modelo de implantación de sistemas se divide luego en un modelo del procesador y uno de tareas.

El modelo del procesador: El diseñador del sistema trata principalmente de decidir cómo se asigna el modelo esencial a los distintos procesadores y como deben comunicarse entre sí.

Existen típicamente una variedad de opciones:

- El modelo esencial completo se le puede asignar a un solo procesador. Esto se conoce como la solución de computadora principal.
- Se puede escoger una combinación de computadoras principales, minis y micros para minimizar costos, maximizar confiabilidad o lograr algún otro objetivo.

Algunas de las opciones disponibles al diseñador del sistema para comunicación de procesador a procesador son:

- Conexión directa entre procesadores. Esto puede implantarse conectando los procesadores mediante un cable, un canal o una red local. Este tipo de comunicación generalmente permite que los datos se transmitan a velocidades de los 50,000 bits por segundo a varios millones de bits por segundo
- Enlace de telecomunicaciones entre procesadores. Esto es común si los procesadores están separados físicamente por algunos cientos de metros. Se transmiten datos a una velocidad desde 300 hasta 50,000 bits por segundo

- Enlace indirecto entre procesadores. Los datos pueden escribirse en cinta magnética, disco flexible, tarjetas perforadas o algún de almacenamiento en un solo procesador y luego ser llevados físicamente al otro para ser empleados como entradas.

El diseñador debe tomar en cuenta varios factores al hacer estas asignaciones, las cuestiones principales son:

- Costo. Dependiendo de la naturaleza del sistema, pudiera ser o no ser más barata una implantación de un solo procesador.
- Eficiencia. El diseñador de sistemas generalmente se preocupa por el tiempo de respuesta de los sistemas en línea y por la longitud del ciclo para los sistemas de cómputo por lote.
- Seguridad. El usuario final podría tener requerimientos de seguridad que dicen que algunos procesadores y/o datos delicados se coloquen en lugares protegidos.
- Confiabilidad. El usuario final típicamente especifica los requerimientos de confiabilidad para un nuevo sistema.
- Restricciones políticas y operacionales. La configuración de hardware puede verse influenciada también por restricciones políticas impuestas directamente por el usuario final.

El modelo de tareas: La noción de tarea es común a casi cualquier marca de hardware de computadora, aunque la terminología difiera de un proveedor a otro: algunos usan el término partición y otros puntos de control. Dentro de un procesador individual, no siempre está claro si las actividades ocurren de manera sincronizada o no; es decir, no siempre queda claro si está sucediendo una cosa o muchas a la vez.

El modelo de implantación de programas: Finalmente se llega al nivel de una tarea individual; hasta aquí el diseñador ya logro completar dos niveles de asignación de procesos y almacenamiento de datos. La computadora opera de una manera no sincronizada: solo se puede llevar a cabo una acción a la vez.

METAS Y OBJETIVOS DEL DISEÑO

La capacidad que los programadores exhiban para implantar un sistema de alta calidad y libre de errores depende en gran medida de la naturaleza del diseño; de manera similar, la capacidad de los programadores de mantenimiento para realizar cambios en el sistema después de haberlo puesto en operación depende de la calidad del diseño. Las dos reglas más importantes son las referentes al acoplamiento y la cohesión:

- Cohesión. Grado en el cual los componentes de un modulo son necesarios y suficientes para llevar a cabo una solo función bien definida.
- Acoplamiento. Grado en el cual los módulos se interconectan o se relacionan entre ellos. En la práctica, esto significa que cada modulo debe tener interfaces sencillas y limpias con otros, y que se debe compartir un numero mínimo de datos entre módulos.

PROGRAMACION Y PRUEBA

EL PAPEL DEL ANALISTA EN LA PROGRAMACION Y LA PRUEBA: Algunas razones por las cuales puede requerir seguir involucrado en el proyecto al comenzar la actividad de programación:

- Usted es el líder del proyecto y está a cargo de los programadores.
- Usted es un analista junior y su titulo es de programador/analista, por ello estará involucrado en la estructura de programas.
- Usted forma parte de un grupo que escribe casos de prueba que se usaran para ejercitar los programadores escritos por los programadores.
- Puede verse involucrado en el desarrollo de manuales de usuario, preparación de los usuarios o en la planeación de la instalación del nuevo sistema y conversión de datos desde el otro sistema.
- Tal vez los programadores no comprendan su especificación. O su especificación puede estar incompleta, ser inconsistente, o ser contradictoria.

EL IMPACTO DEL ANALISIS, LA PROGRAMACION Y LA PRUEBA SOBRE LA ESTRUCTURA ORGANIZACIONAL: El proceso de diseño involucra el desarrollo de modelos de diseño que van desde diagramas de estructura de alto nivel hasta formas de nivel tan bajo como el pseudocódigo y los diagramas de flujo.

Quienes tienen el título de analista usualmente son personas relativamente maduros, con varios años de experiencia. A pesar de que generalmente disfrutan la labor de dibujar diagramas de flujo de datos y diagramas de entidad-relación, les resulta difícil emocionarse ante la perspectiva de tener que escribir cientos de especificaciones de procesos y definir miles de datos.

El personal con experiencia no solo haría las actividades de alto nivel del análisis sino también las actividades de diseño de alto nivel, e incluso llegar a escribir algo de código de alto nivel. La ventaja para los programadores es que les toca hacer el trabajo creativo de escribir las especificaciones de proceso y tienen el placer de traducir sus propias especificaciones a código.

IMPLANTACION DESCENDENTE Y SEGUIMIENTO RAPIDO: ¿Por qué consideraría un administrador del proyecto seguir el enfoque radical? ¿Por qué podría alguien comenzar la labor de diseño y programación antes de concluir la del análisis? Existen muchas razones, de las cuales las más importantes son las siguientes:

- Como la labor de análisis, diseño y programación se realiza de manera concurrente, usualmente se tiene la oportunidad de acortar dramáticamente el tiempo total necesario para un proyecto.
- La labor de desarrollo concurrente puede usarse como una forma de hacer el prototipo.
- El comenzar la labor de programación pronto, suele evitar diversos problemas referentes a la demanda de recursos, tales como tiempo de cómputo, que de otra manera se convertirán en un obstáculo.

PROGRAMACION Y LEGUAJES DE PROGRAMACION

Las cuatro generaciones de lenguajes de programación: Es conveniente agrupar a los distintos lenguajes de programación en cuatro generaciones distintas:

- Lenguaje de primera generación. Fueron los lenguajes máquina que se usaron en los años 50's
- Lenguajes de segunda generación. Son los sucesores del lenguaje máquina; generalmente se conocen como lenguajes de ensamble o ensambladores.
- Lenguajes de tercera generación. Son la norma actual; incluyen BASIC, COBOL, FORTRAN, Pascal, C, Ada y muchos más.
- Lenguajes de cuarta generación. Muchos consultores de computación los consideran el desarrollo más importante en el campo de software en los últimos 20 años.

Cuestiones importantes en programación:

Como analista debe estar familiarizado con ellas; a continuación se mencionan las más comunes:

- Productividad. Probablemente, la cuestión más importante de la programación actual sea la productividad; escribir más software, rápidamente.
- Eficiencia. Minimizar la cantidad de tiempo de CPU requerido por el programa.
- Corrección. Se podría argumentar que esto es lo más importante, después de todo, si el programa no funciona correctamente no importa que tan eficiente sea.
- Portabilidad. El usuario puede desear ejecutar el mismo sistema en varios tipos distintos de computadoras.
- Mantenibilidad. Los sistemas viven durante mucho tiempo por lo que el software debe mantenerse.

Cosas de las que hay que tener cuidado: Como analista puede tener la oportunidad de observar el trabajo que realizan los programadores del proyecto. Algunas cuestiones clave en la programación:

- Programación estructurada. Debe seguirse un enfoque de programación estructurada, en el que la lógica del programa se organiza en combinaciones anidadas de construcciones SI-ENTONCES y HACER-MIENTRAS.
- Módulos pequeños: es esencial que los programas se organicen en pequeños módulos para que la lógica de programación quepa en una sola página de listado de programa.
- Sencillez de estilo. Programas que el programador promedio pueda entender y que se le puedan pasar al programador de mantenimiento.

PRUEBAS

El proceso de probar el sistema toma tanto como la mitad del tiempo programando para su desarrollo, dependiendo de qué tan cuidadosamente se hayan hecho las actividades iniciales de análisis, diseño y programación.

Tipos de pruebas

- Funcional. Su propósito es asegurar que el sistema realiza sus funciones normales de manera correcta.
- Recuperación. Que el sistema pueda recuperarse adecuadamente de distintos tipos de fallas.
- Desempeño. Aseguran que el sistema pueda manejar el volumen de datos y transacciones de entrada especificados en el modelo de implantación del usuario, además de asegurar que tenga el tiempo de respuesta requerido

Información que debe contener un documento de planeación de prueba:

- Propósito de la prueba. Cuál es el objetivo de la prueba, y que parte del sistema se está probando.
- Localización y horario de la prueba. Donde y cuando se hará.
- Descripciones de la prueba. Descripción de las entradas que se proporcionaran al sistema, y las salidas y resultados que se anticipan.
- Procedimientos de prueba. Descripción de cómo se deben preparar y presentar los datos de prueba al sistema.

MANTENIMIENTO DE LA ESPECIFICACION DURANTE LA PROGRAMACION

Es posible que la especificación estructurada cambie durante el proceso de programación. Esto puede suceder como resultado de la estrategia de seguimiento rápido que se describió antes, o porque las especificaciones organizacionales estaban mal, o simplemente porque los usuarios cambian de opinión sobre sus requerimientos.

- Conversión. Tarea de traducir los archivos, formas y bases de datos actuales del usuario al formato que el nuevo sistema requiere. Se necesita desarrollar un plan de conversión, de preferencia en cuanto se complete el modelo de implantación del usuario.
- Instalación. La instalación del nuevo sistema puede ser un asunto instantáneo; pero a menudo es una tarea enorme.
- Capacitación. Es la tarea final del equipo de desarrolladores del sistema: la capacitación de los usuarios, además de la preparación del personal de operaciones, los programadores de mantenimiento y varios niveles de administración.

CONCLUSIONES:

Debemos de tomar en cuenta el modelo de implantación del usuario y mas que nada debemos de entender que lo importante del un sistema es el usuario, el diseño se debe de acoplar a el usuario final, con el fin de hacer mas optimo su trabajo, como bien sabemos en cada etapa del desarrollo del sistema debemos de realizar pruebas a fin de que al ultimo encontremos un producto eficaz.

A la hora de programar debemos realizar por módulos con el fin de que cuando el usuario quiera realizar algún cambio en alguna actividad del sistema podamos realizarlo de manera rápida y sin la necesidad de mover todo el sistema.