TEMA 49

ANÁLISIS DE SISTEMAS: MODELIZACIÓN DE TRATAMIENTOS. MODELO DE FLUJO DE DATOS Y CONTROL. TÉCNICAS DESCRIPTIVAS. DOCUMENTACIÓN.

ÍNDICE

1. INTRODUCCIÓN

El análisis estructurado, como todos los demás métodos de análisis de requisitos, es una actividad de construcción de modelos. Mediante una notación que es única del método de análisis estructurado, creamos modelos que reflejan el flujo y el contenido de la información (datos y control); partimos el sistema funcionalmente y, según los distintos comportamientos, establecemos la esencia de lo que se debe construir. El análisis estructurado no es un método sencillo que se aplica siempre de la misma forma. Más bien, es una amalgama que ha evolucionado durante los últimos 20 años.

Probablemente no exista otro método de ingeniería del software que haya despertado tanto interés, que haya sido probado (y a menudo descartado y vuelto a probar) por tanta gente, que haya provocado tantas críticas y tanta controversia, Sin embargo, el método ha prosperado y cada vez es utilizado más ampliamente dentro de la comunidad de la ingeniería del software.

2. NOTACIÓN BÁSICA Y SUS AMPLIACIONES

A medida que fluye por un sistema basado en ordenador, la información se transforma. El sistema acepta entradas en una gran variedad de formas, aplica los elementos de hardware, software y humanos para transformar la entrada en salida y produce salida en una gran variedad de formas. La entrada puede ser una señal de control transmitida por un controlador, una serie de números escritos en el teclado por un operador humano, un paquete de información transmitido por un enlace de una red o un archivo voluminoso de datos que se encuentra en un dispositivo de almacenamiento secundario. La transformación puede ser, desde una sencilla comparación lógica, hasta un complejo algoritmo numérico o un mecanismo de reglas de inferencia de un sistema experto. La salida puede ser el encendido de un diodo de emisión de luz (LED) o un informe de 200 páginas. Efectivamente, podemos crear un modelo de flujo para cualquier sistema de ordenador, independientemente del tamaño y de la complejidad.

El análisis estructurado es una técnica de modelización del flujo y del contenido de la información. Tal como muestra la Figura, el sistema basado en ordenador se representa como una transformación de información. Se representa el funcionamiento general del sistema como una única transformación de información, que en la figura aparece como una burbuja. En entidades externas, representadas por cuadros, se originan una o más entradas, que aparecen como flechas etiquetadas. La entrada conduce la transformación que produce información de salida (también representada como flechas etiquetadas) dirigida hacia otras entidades externas. Se debe señalar que se puede aplicar el modelo al sistema completo o solamente al elemento de software. La clave está en representar la información que entra y la que es producida por la transformación.

Figura 49.1. Modelo de flujo de información

2.1. Diagramas de flujo de datos

A medida que la información se mueve a través del software, es modificada por una serie de transformaciones. El **diagrama de flujo de datos** (**DFD**) es una técnica gráfica que representa el flujo de la información y las transformaciones que se aplican a los datos al moverse desde la entrada hasta la

salida. En la Figura anterior se muestra la forma básica de un diagrama de flujo de datos. El DFD también es conocido como grafo de flujo de datos o como diagrama de burbujas.

Se puede usar el diagrama de flujo de datos para representar un sistema o un software a cualquier nivel de abstracción. De hecho, los DFDs pueden ser refinados en niveles que representen un mayor flujo de información y un mayor detalle funcional. Un DFD de nivel 0 también es denominado modelo fundamental del sistema o *modelo de contexto*, y representa al elemento de software completo como una sola burbuja con datos de entrada y de salida representados por flechas de entrada y de salida, respectivamente. Al partir el DFD de nivel 0 para mostrar más detalles, aparecen representados procesos (burbujas) y caminos de flujo de información adicionales. Por ejemplo, un DFD de nivel 1 puede contener cinco o seis burbujas con flechas interconectándolas. Cada uno de los procesos representados en el nivel 1 es una subfunción del sistema general en el modelo del contexto.

El rectángulo se usa para representar una entidad externa, es decir, un elemento del sistema (p. ej.: hardware, una persona, otro programa), u otro sistema que produzca información a ser transformada por el software o que reciba información producida por el software. Un círculo representa un proceso o transformación que se aplica a los datos (o al control) y los cambia de alguna forma. Todas las flechas de un diagrama de flujo de datos deben estar etiquetadas. La línea doble representa un almacén de información -información almacenada que es utilizada por el software. La sencillez de la notación DFD es una de las razones por las que las técnicas de análisis estructurado son ampliamente utilizadas.

Es importante señalar que el diagrama no proporciona ninguna indicación explícita de la secuencia de procesamiento. El procedimiento o la secuencia puede estar implícitamente en el diagrama, pero la representación procedimental explícita generalmente queda pospuesta hasta el diseño del software.

Como ya indicamos antes, se puede refinar cada una de las burbujas en distintos niveles para mostrar un mayor detalle. La Figura siguiente ilustra este concepto. El modelo fundamental del sistema F muestra la entrada principal A y la salida final B. Refinamos el modelo F en las transformaciones f1 a f7. Observe que se debe mantener la continuidad del flujo de información, es decir, que la entrada y la salida de cada refinamiento debe ser la misma. Este concepto, a veces denominado equilibrado, es esencial para el desarrollo de modelos consistentes, Un mayor refinamiento de f4 muestra más detalle en la forma de las transformaciones f4l a f45. De nuevo, la entrada (X, Y) y la salida (Z) permanecen inalteradas.

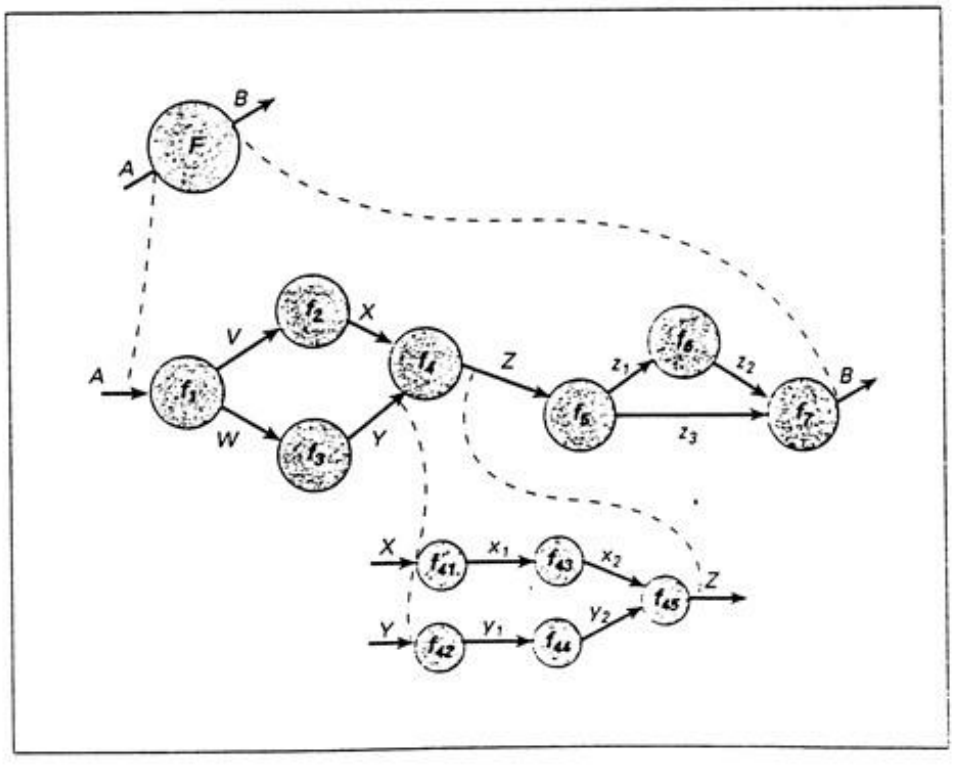

Figura 49.2. Refinamiento del flujo de información

El diagrama de flujo de datos es una herramienta gráfica que puede ser muy valiosa durante el análisis de requisitos del software. Sin embargo, el diagrama puede llevar a confusión si se confunde su función con la del diagrama de flujo. Un diagrama de flujo de datos representa el flujo de la información sin representación explícita de la lógica de procesamiento (p. ej.: condiciones o bucles). ¡No se trata de un diagrama de flujo con elementos redondeados!.

La notación básica que se usa para desarrollar un DFD no es en sí misma suficiente para describir los requisitos del software. Por ejemplo, una flecha de un DFD representa un elemento de datos que entra o sale de un proceso. Un almacén de datos representa alguna colección organizada de datos. Pero, ¿cuál es el contenido de los datos implicados en las flechas o en el almacén? Si la flecha (o el almacén) representan una colección de elementos, ¿cuáles son? Para responder a estas preguntas, aplicamos otro componente de la notación básica del análisis estructurado -el diccionario de requisitos, también denominado **diccionario de datos**. El formato y el uso del diccionario de requisitos se explican más adelante.

Finalmente, la notación gráfica debe ser ampliada con texto descriptivo. Se puede usar una narrativa de procesamiento -un párrafo que describe una burbuja de procesamiento- para especificar los detalles de procesamiento que implica una burbuja del DFD. La narrativa de procesamiento describe la entrada a la burbuja, el algoritmo que se aplica a esa entrada y la salida que se produce. Además, la narrativa indica las restricciones y limitaciones impuestas al proceso, las características de rendimiento que son relevantes al proceso y las restricciones de diseño que puedan tener influencia en la forma de implementar el proceso.

2.2. Modelización del comportamiento

La modelización del comportamiento es uno de los principios fundamentales de todos los métodos de análisis de requisitos. Sin embargo, sólo algunas versiones ampliadas del análisis estructurado proporcionan una notación para este tipo de modelización. El **diagrama de transición de estados** representa el comportamiento de un sistema que muestra los estados y los sucesos que hacen que el sistema cambie de estado. Además, el **DTE** indica qué acciones (p. ej.: activación de procesos) se llevan a cabo como consecuencia de un suceso determinado.

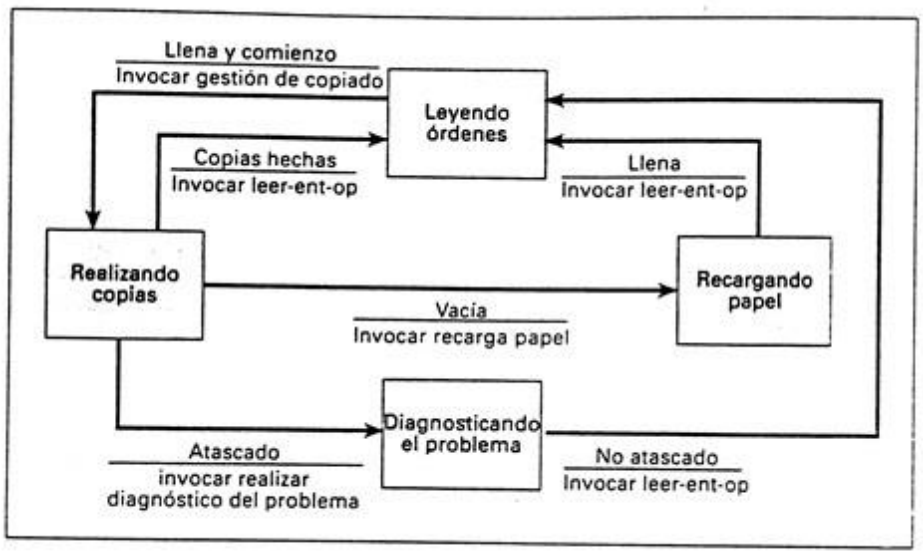

Figura 49.3. Diagrama de transición de estados simplificados para el software de fotocopiadora

En la Figura se muestra un diagrama de transición de estados simplificado para el software de una fotocopiadora descrito en la sección anterior. Los rectángulos representan estados del sistema y las flechas representan transiciones entre estados. Cada flecha está etiquetada con una expresión en forma de fracción. La parte superior indica el suceso (o sucesos) que hace(n) que se produzca la transición. La parte inferior indica la acción que se produce como consecuencia del suceso. Así, cuando la bandeja de papel

está llena, y el botón de comienzo ha sido pulsado, el sistema pasa del estado leyendo órdenes al estado realizando copias. Observe que los estados no se corresponden necesariamente con los procesos de forma biunívoca. Por ejemplo, el estado realizando copias englobaría tanto el proceso gestión de copiado como el proceso producir visualización de usuario.

3. LA MECÁNICA DEL ANÁLISIS ESTRUCTURADO

En la sección anterior, hemos visto las notaciones básica y ampliada del análisis estructurado. Para poder utilizarlas eficientemente en el análisis de requisitos del software, se ha de combinar esa notación con un *conjunto de heurísticas* que permitan al ingeniero del software derivar un buen modelo de análisis. Para ilustrar el uso de esas heurísticas, en el resto de este capítulo utilizaremos una versión adaptada de las ampliaciones de Hatley y Pirbhai a la notación básica del análisis estructurado.

En las secciones siguientes, se examina cada uno de los pasos que se deben seguir para desarrollar modelos completos y precisos mediante el análisis estructurado.

3.1. Creación de un Modelo de Flujo de Datos

El diagrama de flujo de datos (DFD) permite al ingeniero de software desarrollar los modelos del ámbito de información y del ámbito funcional al mismo tiempo. A medida que se refina el DFD en mayores niveles de detalle, el analista lleva a cabo implícitamente una descomposición funcional del sistema. Al mismo tiempo, el refinamiento del DFD produce un refinamiento de los datos a medida que se mueven a través de los procesos que componen la aplicación.

Unas pocas *directrices* sencillas pueden ayudar de forma considerable durante la derivación de un diagrama de flujo de datos: **(1)** el diagrama de flujo de datos de nivel 0 debe reflejar el software/sistema como una sola burbuja; **(2)** se deben anotar cuidadosamente la entrada y la salida principales; **(3)** el refinamiento debe comenzar aislando los procesos, los elementos de datos y los almacenes de datos que sean candidatos a ser representados en el siguiente nivel; **(4)** todas las flechas y las burbujas deben ser rotuladas con nombres significativos; **(5)** entre sucesivos niveles se debe mantener la continuidad del flujo de información; **(6)** se deben refinar las burbujas de una en una. Hay una tendencia natural a complicar en exceso el diagrama de flujo de datos. Esto ocurre cuando el analista intenta reflejar demasiados detalles demasiado pronto o representa aspectos procedimentales en detrimento del flujo de información. Para ilustrar el uso de estas directrices básicas, usaremos un *ejemplo* de un sistema de seguridad, que denominaremos HogarSeguro. A continuación, describimos el procesamiento para HogarSeguro:

(El software HogarSeguro permite al propietario de la vivienda configurar el sistema, de seguridad al instalarlo; controla todos los sensores conectados al sistema de seguridad e interactúa con el propietario a través de un teclado numérico y unas teclas de función que se encuentran en el panel de control de HogarSeguro).

Durante la instalación, se usa el panel de control de HogarSeguro para "programar" y configurar el sistema. Cada sensor tiene asignado un número y un tipo; existe una contraseña maestra para activar y desactivar el sistema, y se introduce(n) un(os) teléfono(s) con los que contactar cuando se produce un suceso detectado por un sensor.

Cuando el software detecta la sensorización de un suceso, hace que suene una alarma audible que está incorporada en el sistema. Tras un retardo, especificado por el propietario durante la configuración del sistema, el programa marca un número de teléfono de un servicio de monitorización, proporciona información sobre la situación e informa sobre la naturaleza del suceso detectado. Cada 20 segundos se volverá a marcar el número de teléfono hasta que se consiga establecer la comunicación.

Toda la interacción con HogarSeguro está gestionada por un subsistema de interacción con el usuario que lee la información introducida a través del teclado numérico y las teclas de función, muestra mensajes de petición en un monitor LCD y muestra información sobre el estado del sistema en el monitor LCD.

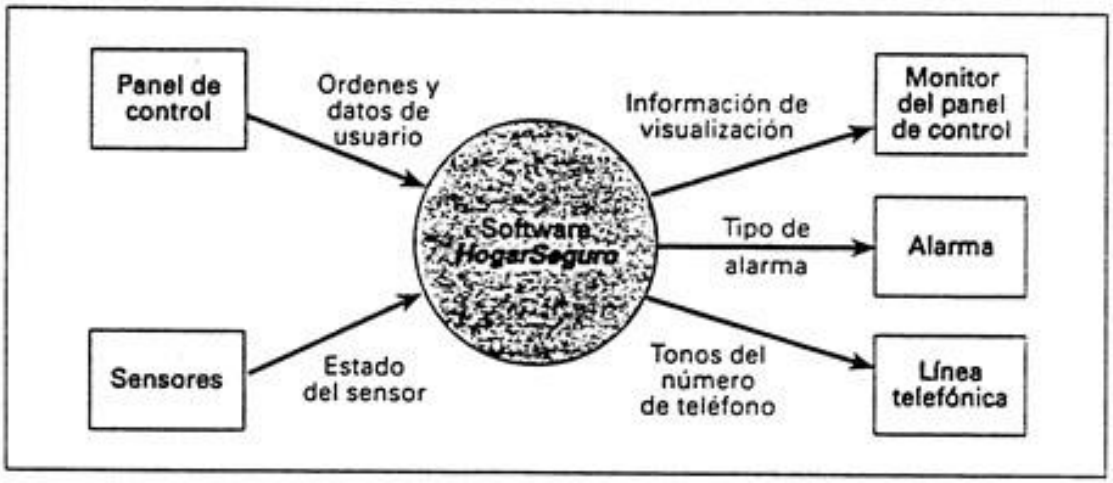

Figura 49.4. DFD de nivel contextual para HogarSeguro

En la Figura se muestra el DFD de nivel 0 para HogarSeguro. Las entidades externas principales (cuadros) producen información a ser usada por el sistema y consumen información generada por el sistema. Las flechas etiquetadas representan elementos de datos compuestos, es decir, elementos de datos que consisten realmente en colecciones de otros muchos elementos de datos adicionales. Por ejemplo, órdenes y datos de usuario engloban todas las órdenes de configuración, todas las órdenes de activación/desactivación, todas las variadas interacciones y todos los datos que se introducen para cualificar o ampliar una orden.

Ahora, tenemos que expandir el DFD de nivel 0 a un modelo de nivel 1. Pero, ¿cómo lo hacemos? Una sencilla, pero efectiva, técnica consiste en realizar un "análisis gramatical" de la narrativa de procesamiento que describe la burbuja de nivel contextual. Es decir, aislar todos los nombres (y sentencias nominales) y todos los verbos (y sentencias verbales) de la narrativa presentada arriba. Para ilustrarlo, reproducimos de nuevo la narrativa de procesamiento, subrayando las primeras ocurrencias de los nombres y con las primeras ocurrencias de los verbos en cursiva.

Se debe tener en cuenta que se ignoran los nombres y los verbos que sean sinónimos o que no incumban directamente al proceso de modelización.

De acuerdo con el análisis gramatical, comienza a aparecer un patrón. Todos los verbos son procesos de HogarSeguro; es decir, deben estar en última instancia representados como burbujas en el consiguiente DFD. Todos los nombres son, o bien entidades externas (cuadros), elementos de datos o de control (flechas) o almacenes de datos (líneas dobles). Además, los nombres y los verbos están relacionados entre sí (p. ej.: cada sensor tiene asignado un número y un tipo). Por tanto, con el análisis gramatical de la narrativa de procesamiento de una burbuja de cualquier nivel de DFD, podemos generar mucha información útil sobre cómo proceder en el refinamiento del siguiente nivel. Usando esa información, obtenemos el DFD de nivel 1 que se muestra en la Figura 5. El proceso del nivel contextual de la Figura 4 se ha expandido en siete procesos, derivados de un examen del análisis gramatical. De forma similar se ha derivado el flujo de información entre los procesos del nivel 1.

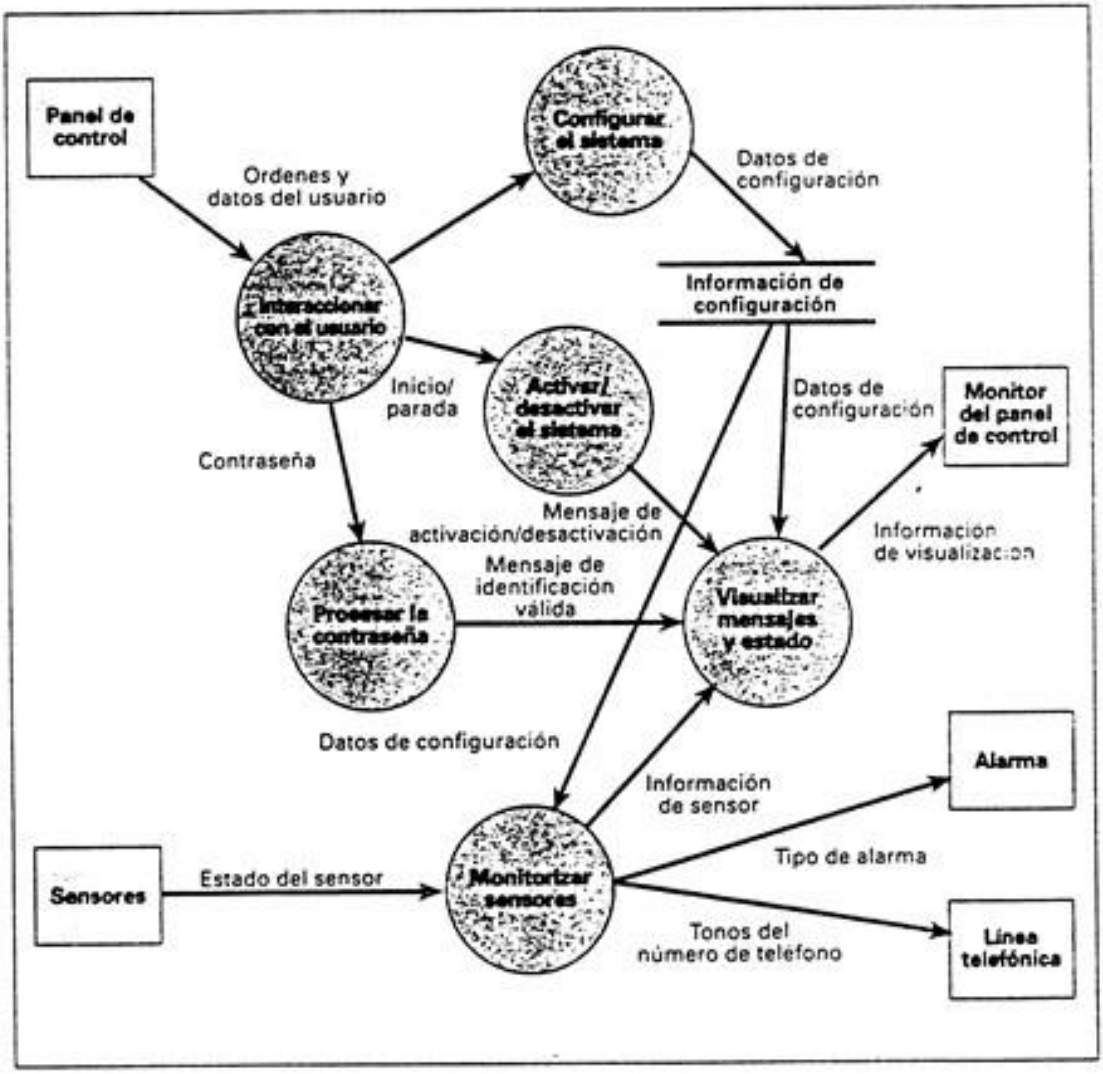

Figura 49.5. DFD de nivel 1 para HogarSeguro

Se debe tener en cuenta que entre los niveles 0 y 1 se ha mantenido la continuidad del flujo de información.

Se pueden refinar en posteriores niveles los procesos representados en el DFD de nivel 1. Por ejemplo, se puede refinar el proceso monitorizar sensores al DFD de nivel 2 que se muestra en la Figura 6. Podemos observar de nuevo que se mantiene la continuidad del flujo de información entre los niveles. El refinamiento de los DFDs continúa hasta que cada burbuja representa una función sencilla, es decir, hasta que el proceso que representa la burbuja realiza una función que se puede implementar fácilmente como una componente del programa.

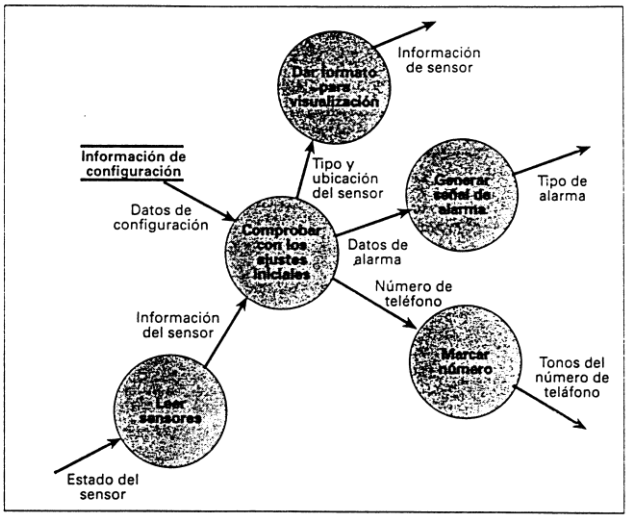

Figura 49.6. DFD de nivel 2 que refina el proceso monitorizar sensores

Durante el análisis de requisitos, el ingeniero del software puede detectar ciertos aspectos del sistema que "son susceptibles al cambio" o que "serán mejorados en el futuro" o que han sido definidos por el cliente de forma poco clara. También puede ser que el analista esté trabajando sobre un software existente que va a ser posteriormente modificado. En cualquiera de estos casos, el diagrama de flujo de datos permite aislar fácilmente el campo de cambio. Comprendiendo perfectamente el flujo de información entre los límites del campo de cambio, se pueden preparar mejor las futuras modificaciones, o las modificaciones que se han de llevar a cabo en ese momento sin que afecten a otros elementos del sistema.

3.2. Creación de un Modelo de Flujo de Control

Para muchos tipos de aplicaciones de procesamiento de datos, todo lo que se necesita para obtener una representación significativa de los requisitos del software es el modelo de flujo de datos. Sin embargo, como ya hemos mencionado anteriormente, existe una clase de numerosas aplicaciones que están "conducidas" por sucesos en lugar de por los datos, que producen información de control más que informes o visualizaciones, que procesan información con fuertes limitaciones de tiempo y rendimiento. Tales aplicaciones requieren una modelización del flujo de control además de la modelización del flujo de información.

Para crear un **DFC**, lo primero es "eliminar" del modelo de flujo de datos todas las flechas de flujo de datos. Luego, se añaden al diagrama los sucesos y los elementos de control (flechas con línea discontinua), así como "ventanas" (barras verticales) a las especificaciones de control. Pero, ¿cómo se seleccionan los sucesos? Ya hemos señalado que los sucesos o los elementos de control se implementan como valores lógicos (p. ej.: verdadero o falso, si o no, 1 o 0) o como una lista discreta de condiciones (vacía, atascada, llena). Para seleccionar posibles candidatos a sucesos, se pueden sugerir las siguientes directrices:

Listar todos los sensores que son "leídos" por el software.
Listar todas las condiciones de interrupción.
Listar todos los "interruptores" que son accionados por el operador.
Listar todas las condiciones de datos.
De acuerdo con el análisis de nombres y verbos que se aplicó a la narrativa de procesamiento, revisar todos los "elementos de control" como posibles entradas/salidas de CSPECs (**CSPEC: Especificación de Control**).
Describir el comportamiento del sistema identificando sus estados; identificar cómo se alcanza cada estado y definir las transiciones entre los estados.
Centrarse en las posibles omisiones (un error muy común en la especificación del control); por ejemplo, preguntarse si existe alguna otra forma en la que se puede llegar a un estado o salir de él.

En la Figura se ilustra un DFC de nivel 1 para el software HogarSeguro. Entre los sucesos y los elementos de control que aparecen están el suceso de sensor (p. ej.: un sensor ha detectado una anomalía), indicador de parpadeo (una señal para que el monitor LCD parpadee) y el interruptor de comienzo/parada (una señal para encender y apagar el sistema). Cuando un suceso fluye a una barra CSPEC desde el mundo exterior, ello implica la activación por la CSPEC de uno o más procesos de los que aparecen en el DFC. Cuando un elemento de control emana de un proceso y fluye a una ventana CSPEC, ello implica el control y la activación de algún proceso o de una entidad externa.

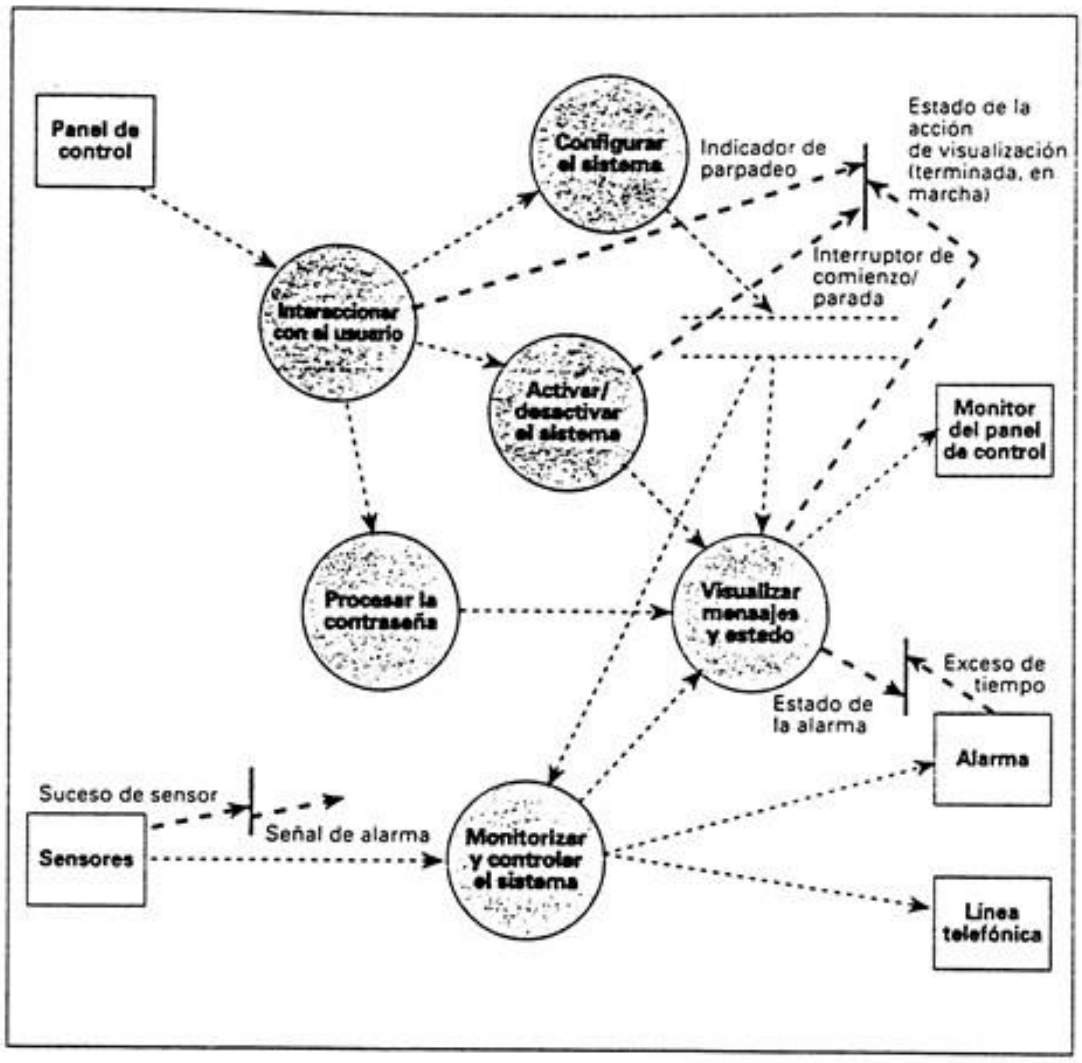

Figura 49.7. DFC de nivel 1 para HogarSeguro

3.3. La Especificación de Control (CSPEC)

La especificación de control representa el comportamiento del sistema (al nivel al que está referenciada) de dos formas diferentes. La CSPEC contiene un **diagrama de transición de estados** (**DTE**) que es una especificación secuencial del comportamiento. También puede contener una **tabla de activación de procesos** (**TAP**) (una especificación combinatoria del comportamiento). Ahora es el momento de examinar un *ejemplo* de esta importante notación de modelización del análisis estructurado.

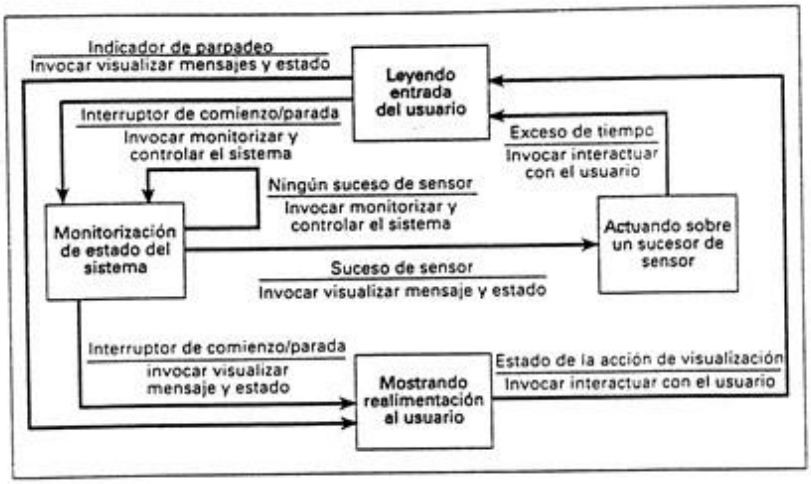

Figura 49.8. Diagrama de transición de estados para HogarSeguro

La Figura refleja un diagrama de transición de estados para el modelo de flujo de nivel 1 de HogarSeguro. Las flechas de transición etiquetadas indican cómo responde el sistema a los sucesos a medida que pasa por los cuatro estados definidos en este nivel. Estudiando el DTE, un ingeniero del software puede determinar el comportamiento del sistema y, lo que es más importante, comprobar si hay "lagunas" en el comportamiento especificado. Por ejemplo, el DTE indica que sólo se produce una transición desde el estado "leyendo entrada del usuario" cuando se encuentra interruptor de comienzo/parada, produciéndose una transición al estado "monitorizando el estado del sistema". En efecto, no parece haber ninguna forma, aparte de la ocurrencia de suceso de sensor o de interruptor de comienzo/parada, que permita al sistema volver al estado "leyendo entrada del usuario". Esto puede ser un error de la especificación y, desgraciadamente, podría no haber sido detectado y corregido durante la revisión.

Una representación algo diferente del modo de comportamiento es la *tabla de activación de procesos*. La TAP representa la información contenida en el DTE dentro del contexto de los procesos, no de los estados. Es decir, la tabla indica los procesos (burbujas) del modelo de flujo que serán invocados cuando se produzca un suceso. Se puede usar la TAP como una guía para el diseñador que tenga que construir un ejecutor que controle los procesos representados en ese nivel. En la Figura se muestra una TAP para el modelo de flujo de nivel 1 de HogarSeguro.

Sucesos de entrada	
Suceso de sensor	000010
Indicador de parpadeo	001100
Interruptor de comienzo/parada	010000
Estado de la acción de visualización	
Terminada	000100
En marcha	001000
Exceso de tiempo	000001
Salida	
Señal de alarma	000000
Activación de procesos	
Monitorizar y controlar el sistema	010011
Activar/desactivar el sistema	000000
Visualizar mensajes y estado	111111
Interactuar con el usuario	100101

Figura 49.9. Tabla de Activación de procesos para HogarSeguro

La CSPEC describe el comportamiento del sistema, pero no nos proporciona información sobre el funcionamiento interno de los procesos que son activados como resultado de ese comportamiento. En la siguiente sección se discute la notación de modelización que proporciona esa información.

3.4. La Especificación de Procesamiento

Se usa la **especificación de procesamiento** (**PSPEC**) para describir todos los procesos del modelo de flujo que aparecen en el nivel final de refinamiento. El contenido de la especificación de procesamiento puede incluir una narrativa textual, una descripción en **lenguaje de descripción de programa** (**LDP**) del algoritmo del proceso, ecuaciones matemáticas, tablas, diagramas o gráficos. Al proporcionar una PSPEC que acompañe cada burbuja del modelo de flujo, el ingeniero del software crea una "mini-especificación" que sirve como primer paso para la creación de la especificación de requisitos del software y constituye una guía para el diseño de la componente de programa que implementará el proceso.

Para ilustrar el uso de la PSPEC, consideremos una aplicación de software en la que se analizan las dimensiones de varios objetos geométricos para identificar la forma del objeto. Se lleva a cabo el refinamiento del diagrama de flujo de datos de nivel contextual hasta que se derivan los procesos del nivel 2. Uno de ellos, denominado analizar triángulo, aparece reflejado en la Figura. Primero se escribe una narrativa de la PSPEC de analizar triángulo en castellano tal como muestra la figura. Si en esta etapa se desea incluir detalles algorítmicos adicionales, se puede incluir como parte de la PSPEC su representación

en lenguaje de descripción de programa. Sin embargo, muchos piensan que se debe posponer la versión en LDP hasta que comience el diseño.

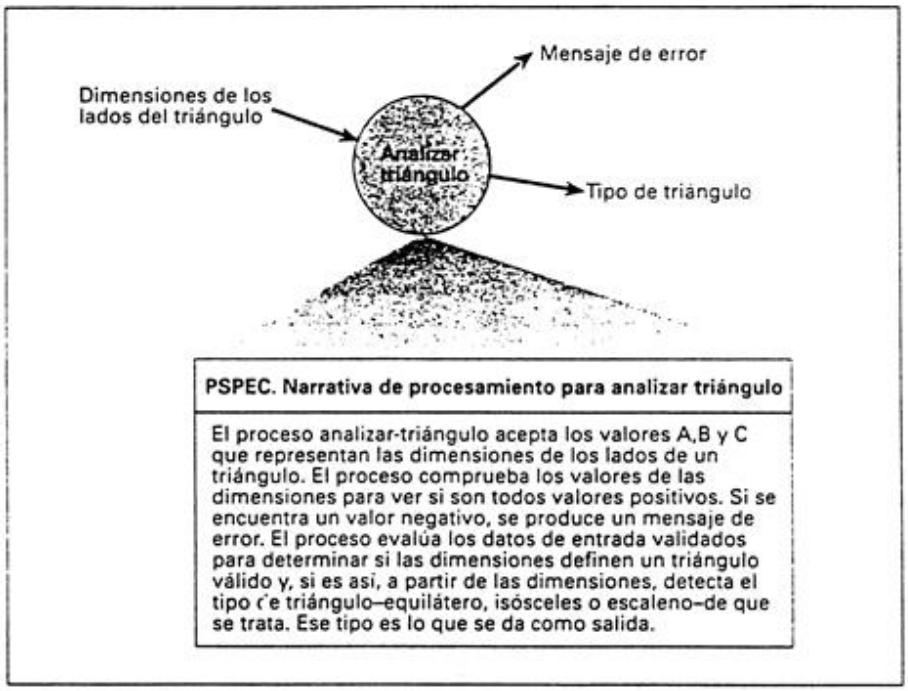

Figura 49.10. Especificación de procesamiento

4. EL DICCIONARIO DE REQUISITOS

Un análisis del ámbito de información estaría incompleto si sólo se considera el flujo de la información. Cada flecha del diagrama de flujo de datos representa un elemento de información o varios. Cada almacén de información a menudo es una colección de elementos de datos individuales. Puede que cada elemento de control esté definido en términos de otros elementos de control. Incluso puede que el contenido de una entidad externa requiera ser expandido antes de que su significado pueda ser definido explícitamente. Por tanto, el analista debe disponer de algún método para representar el contenido de cada componente del modelo de flujo.

Se ha propuesto el **diccionario de requisitos** (también denominado *diccionario de datos*) como gramática casi formal para describir el contenido de los objetos definidos durante el análisis estructurado. Esta importante notación de modelización ha sido definida de la siguiente forma:

El **diccionario de datos** es un listado organizado de todos los elementos de datos que son pertinentes para el sistema, con definiciones precisas y rigurosas que permiten que el usuario y el analista del sistema tengan una misma comprensión de las entradas, de las salidas, de las componentes de los almacenes y [también] de los cálculos intermedios.

Actualmente, casi siempre se implementa el diccionario de requisitos como parte de una "herramienta CASE de análisis y diseño estructurados". Aunque el formato del diccionario varía entre las distintas herramientas, la mayoría contiene la siguiente información:

Nombre: el nombre principal del elemento de datos o de control, del almacén de datos o de una entidad externa.
Alias: otros nombres usados para la entrada.
Dónde se usa/cómo se usa: un listado de los procesos que usan el elemento de datos o de control y cómo lo usan (p. ej.: como entrada al proceso, como salida del proceso, como almacén de datos, como entidad externa).
Descripción del contenido: el contenido representado mediante una notación.

Información adicional: otra información sobre los tipos de datos, los valores implícitos (si se conocen), las restricciones o limitaciones, etc.

Una vez que se introducen en el diccionario de requisitos un nombre y sus alias, se debe revisar la consistencia de las denominaciones. Es decir, si un equipo de análisis decide denominar un elemento de datos recién derivado como xyz, pero en el diccionario ya existe otro llamado xyz, la herramienta **Case** que soporta el diccionario muestra un mensaje de alerta sobre la duplicidad de nombres. Esto mejora la consistencia del modelo de análisis y ayuda a reducir errores.

La información de "dónde se usa/cómo se usa" se registra automáticamente a partir de los modelos de flujo. Cuando se crea una entrada del diccionario, la herramienta **Case** inspecciona los DFDs y los DFCs para determinar los procesos que usan el dato o la información de control y cómo lo usan. Aunque esto pueda no parecer importante, realmente es una de las mayores ventajas del diccionario. Durante el análisis, hay una corriente casi continua de cambios. Para proyectos grandes, a menudo es bastante difícil determinar el *im*pacto de un cambio. Algunas de las preguntas que se plantea el ingeniero de software son "¿dónde se usa este elemento de datos? ¿qué más hay que cambiar si lo modificamos? ¿cuál será el impacto general del cambio?". Al poder tratar el diccionario de requisitos como una base de datos, el analista puede hacer preguntas basadas en dónde se usa/cómo se usa y obtener respuestas a peticiones similares a las anteriores. La notación usada para desarrollar una descripción del contenido permite al analista representar los datos compuestos en una de tres formas fundamentales de construcción:

1. Como una secuencia de elementos de datos.
2. Como una selección de entre un conjunto de elementos de datos.
3. Como una agrupación repetitiva de elementos de datos.

Cada entrada de elemento de datos que aparezca como parte de una secuencia, una selección o una repetición puede a su vez ser otro elemento de datos compuestos que necesite un mayor refinamiento en el diccionario.

Para ilustrar el uso del diccionario de requisitos y la notación de descripción de contenido volvamos al DFD de nivel 2 del proceso monitorizar el sistema de HogarSeguro. Refiriéndonos a la figura, especificamos el elemento de datos número de teléfono. ¿Qué es exactamente un número de teléfono? Puede ser un número local de siete dígitos, una extensión de cuatro dígitos o una secuencia de 14 dígitos para llamadas de larga distancia. El diccionario de requisitos nos proporciona una definición precisa de número de teléfono para el DFD en cuestión. Además, indica dónde y cómo se usa este elemento de datos y cualquier información adicional que le sea relevante. La entrada del diccionario de requisitos comienza de la siguiente forma:

nombre:	número de teléfono
alias:	ninguno
dónde se usa/	
cómo se usa:	Comprobar con ajustes iniciales (salida). Marcar número (entrada)
descripción:	número de teléfono = [extensión local | número exterior]

La descripción de contenido anterior se debe leer como: número de teléfono está compuesto o bien por una extensión local (que se usa en grandes compañías) o por un número exterior. La extensión local y el número exterior son datos compuestos y deben ser refinados en otras sentencias de descripción de contenido. Continuando con la descripción de contenido:

número de teléfono = [extensión local | número exterior]
extensión local = [2001 | 2002 | ... | 2999]
número exterior = (0) + [número local | número de larga distancia]
número local = prefijo + número de acceso
número de larga distancia = código de área + número local
prefijo = [725 | 474 | 255 | 394]
número de acceso = *cualquier cadena de cuatro dígitos*

La descripción de contenido es expandida hasta que se hayan representado como datos elementales (elementos que no requieren más expansión) todos los elementos de datos compuestos o hasta que todos los elementos compuestos aparecen representados en términos bien conocidos y sin ambigüedad para

cualquier lector (p. ej.: todo el mundo sabe que el código de área está compuesto por 2 o 3 dígitos y empieza por un 9). También es importante tener en cuenta que una especificación de un dato elemental a menudo restringe el sistema. Por ejemplo, la definición de prefijo indica que sólo se pueden acceder localmente centralitas de cuatro ramificaciones.

El diccionario de datos define sin ambigüedad los elementos de datos. Aunque podemos asumir que el número de teléfono representado por el DFD puede contener un número de acceso de larga distancia de 14 dígitos, la descripción del contenido del diccionario de requisitos nos indica que esos números no son parte de los datos que se van a usar.

Para grandes sistemas basados en ordenador, el diccionario de requisitos crece rápidamente en tamaño y en complejidad. De hecho, es extremadamente difícil mantener manualmente el diccionario. Por esta razón, se deben usar herramientas **Case.**

5. BIBLIOGRAFÍA

Pressman, Roger S.
Ingeniería del Software
Mc Graw-Hill, 1993

López, Antonio
Metodologías de Desarrollo
Ra-ma, 1990